NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

STUDY OF A POTENTIAL SINGLE POINT HOUSEHOLD COMMUNICATIONS PRODUCT UTILIZING INTERNET PROTOCOL

by

Donna L. Fortin

December 2000

Thesis Advisor:	Gilbert M. Lundy
Second Reader:	James B. Michael

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Study of a Potential Single Point Household Communications Product Utilizing Internet Protocol

5. FUNDING NUMBERS

6. AUTHOR(S) Donna L. Fortin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(maximum 200 words)***

The future of networking technology and the Internet offer a great deal of promise. The potential is forthcoming as newer hardware technology and higher bandwidth capable protocols are designed and implemented. This thesis investigates the possibility of utilizing existing hardware with presently available software to create a practical communication package for the household. The household communication package or home communicator is the network core of the household linking television, telephone, and web browsing capability into one system. The home communicator would receive an incoming television, telephone and Internet signal via optical fiber from a single service provider.

This thesis investigates Linux as the home communicator operating system with Internet Protocol version 6 as the network protocol. Linux is examined for its proficiency at being a capable customer oriented operating system. Additional Linux compatible applications are studied to include web browsing, e-mail, chat and simple text editing. Finally, IPv6 is compared to IPv4 for possible enhancement of the home communicator. Individual aspects of IPv6 are investigated for additional security and better bandwidth. Linux with IPv6 was found to be an acceptable software package for the home communicator. There are several major issues preventing an easy solution. A portion of the functionality must be attained through the Internet Service Provider.

14. SUBJECT TERMS Linux, Internet Protocol, IPv6

15. NUMBER OF PAGES 96

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

STUDY OF A POTENTIAL SINGLE POINT HOUSEHOLD COMMUNICATIONS PRODUCT UTILIZING INTERNET PROTOCOL

Donna L. Fortin
B.S.E.E., Worcester Polytechnic Institute, 1985

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
SOFTWARE ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL
December 2000**

Author: ______________________
Donna L. Fortin

Approved by: ______________________
Gilbert Lundy, Thesis Advisor

James Bret Michael
____
James Michael, Second Reader

____
Luqi, Chair
Software Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The future of networking technology and the Internet offer a great deal of promise. The potential is forthcoming as newer hardware technology and higher bandwidth capable protocols are designed and implemented. This thesis investigates the possibility of utilizing existing hardware with presently available software to create a practical communication package for the household. The household communication package or home communicator is the network core of the household linking television, telephone, and web browsing capability into one system. The home communicator would receive an incoming television, telephone and Internet signal via optical fiber from a single service provider.

This thesis investigates Linux as the home communicator operating system with Internet Protocol version 6 as the network protocol. Linux is examined for its proficiency at being a capable customer oriented operating system. Additional Linux compatible applications are studied to include web browsing, e-mail, chat and simple text editing. Finally, IPv6 is compared to IPv4 for possible enhancement of the home communicator. Individual aspects of IPv6 are investigated for additional security and better bandwidth. Linux with IPv6 was found to be an acceptable software package for the home communicator. There are several major issues preventing an easy solution. A portion of the functionality must be attained through the Internet Service Provider.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION

This thesis describes the study of a new type of household communicator designed to work in a high bandwidth network environment. This is based on the assumption that in the future high speed optical fiber will be available to the average household. The introduction describes the evolution of networking and optical fiber influence on the past and present.

A. FUTURE OPTICAL FIBER NETWORKS

Information Technology is the objective of the future. Since the mid 1980's workstation computers have continued to expand into the work and home environments. In order for businesses to draw on the growing information technology networks have been created to tie the workstations computers together fostering an environment for their employees to easily share applications and information. No longer is it necessary to physically draw up a brief and hand it to the boss. It now can be electronically developed within minutes and delivered within seconds. This is the just the entryway to the power possible from computers and computer networking.

Although home computing has not kept up to the same level of business computing the home market has been expanding as well. Many families now have a network of computers for adult and child interaction. One great advantage to a network environment within the home is that only one ISP connection allows all computers to access the internet at the same time.

The actual computer is just one facet of this growing information technology, though. In order for the computer to work well in a networked environment it is important to have bandwidth and speed between the separate workstations. With the onset of faster more capable computer processors one of the major dilemmas that has developed is increasing the bandwidth and speed of the networked connections in order to keep up with computer speeds. The bandwidth and speed of a network connection is contingent upon two things, the medium and the network protocol utilized. Both of these perform together to produce the actual throughput speeds and bandwidth capability actualized in a network.

The three common guided transmission mediums are twisted pair, coaxial cable, and optical fiber. The general information for bandwidth, total data rate capabilities, and repeater spacing requirements for these mediums are listed in Table 1-1. The repeater spacing requirements for twisted pair and coaxial cable compared to the optical fiber precludes twisted pair and coaxial cable as good procedurally for long distance communications. Twisted pair, CATV, and coaxial cable are the most common within the office building environment due to their lower hardware and installation costs. Optical fiber has been used most often for long distance and metropolitan trunk lines due to its exceptional bandwidth capabilities and repeater spacing distance. Wireless Transmission is not within the scope of this thesis.

Transmission medium	Data Rate Ranges	Bandwidth	Repeater spacing
Twisted pair / Category 3/4/5	64 Kbps – 1 Gbps*	3.5 Kbps - 100 MHz	2 km
Coaxial cable	40 - 500 Mbps	1 – 500 MHz	1 to 10 km**
Optical fiber	2 Gbps – 1 Tbps	2 GHz – 1 THz	10 to 100 km

* Note: data rates above 10 Mbps are limited in terms of the number of devices and geographic scope of the network.

** Note: close spacing for repeaters is required for higher data rates.

Table 1-1 Point –to –point transmission characteristics of guided media

Up through the mid 1990s it was less important to have the high data rate capability down to the workstation computer. Network congestion mostly appeared at the higher levels of the networks rather than at the workstation computer itself. Packets of information transversing the internet consisted mostly of data packets. Today it is more common to transfer large imagery files, voice files, and movie files in addition to the normal smaller data packets between computers in a LAN or over the internet. Also, Voice over IP (VoIP) and video teleconferencing over the internet are much more prevalent today than 5 years ago. VoIP not only require a larger bandwidth to the desktop for proper utilization but they also add timing requirements not found in simple data transfer. It is important that live voice and video be sent within a relatively short period of time with no or few lost packets. Lowered bandwidth during a video teleconferencing can extremely affect video quality. Optical fiber holds the key for greater capabilities.

In the past ATM (Asynchronous Transfer Mode) technology was often used to realize optical fiber bandwidth potential. ATM afforded good throughput for communications, but it came at a high price. The added speed to the network was far outweighed by the ATM hardware cost. Also, by the time an ATM desktop standard was engineered the development of LAN switching technology had taken hold. The time saved in switch utilization within the LAN translated into increased throughput previously lost to non-ATM LANs such as Ethernet, FDDI, and Token Ring. Hence, ATM was prevented from reaching the standard desktop [Ref. 1]. ATM technology settled comfortably into the high bandwidth requirement of backbone LAN to LAN position.

Ethernet protocol technology has increased dramatically in the past few years to further dissuade the use of ATM in networking environments. This new technology has taken the Ethernet speeds up to the Gigabit level. Gigabit Ethernet can easily make use of the optical fiber medium through the IEEE 802.3z standard with 1000BASE-SX and 1000BASE-LX specifications [Ref. 2]. In a Gigabit Ethernet network the interface equipment to the optical fiber medium is more expensive than standard Ethernet technology, but the extended throughput without overall network redesign required with

ATM is inviting. Also, due to the growing technology with Ethernet capability Ethernet now spans all communications media where ATM is limited to optical fiber.

Optical fiber has always been known as a high bandwidth, low error rate medium for communications. The available bandwidth in optical fiber is approximately 50 THz. A peak electronic speed of only a few gigabits per second has previously prevented utilization of optical fiber's inherent higher bandwidth capabilities, though. Very recently a new form of technology, wavelength-division multiplexing (WDM), further extends the practical bandwidth of optical fiber in communications systems.

WDM networks are third-generation technology compared to non-fiber technology utilizing Ethernet and fiber technology utilizing fiber distributed data interface (FDDI) or ATM. Optical fiber that was laid years ago with a calculated second-generation bandwidth cap of 2 Gbps now can be employed with WDM technology creating a considerably greater capacity of more than 40 Gbps. [Ref. 3] WDM technology uses multiple wavelengths of light simultaneously to create channels that operate independently of each other yet within the same physical fiber. Each of these channels can be used to transmit data at the low Gbps rate.

In 1996 researchers at Nippon Telegraph and Telephone (NTT) demonstrated a WDM system that multiplexed 16 separate channels of independent 6.3 Gbps signals into an overall 400 Gbps signal. The WDM multiplexed signal was amplified and transmitted over a 100 km optical fiber transmission line. At the other end this 400 Gbps signal was de-multiplexed into the original individual 16 channels and converted to electrical signals. This experiment was completed within a controlled environment, yet that does not preclude the chances of technology soon leading us to this capability within the existing World Wide Web atmosphere. [Ref. 4]

With the increased utilization of optical fiber capabilities it is obvious the next bottleneck to communications flow lies in the electronic buffer requirement at each network hop. Presently existing networks require electrical processing to be done at every node. The optical fiber technologies thus described are only utilized for point-to-point transmission. It is anticipated that the next generation will utilize more photonic exchanges. Future photonic networks will use direct optical connections without electrical processing. Also, optical switches are presently being developed that handle 320 Gbps [Ref. 4]. The future of optical fiber within the networking arena is highly favored.

Probably one of the strongest indications that optical fiber will have continued presence and growth in the future as a main structure of internet networking is the present business trend to utilize fiber within and between the major industrial countries of the world. Denver based Qwest has built a two billion dollar nationwide fiber-optic network as a backbone to its telephone service [Ref. 5]. Alcatel and Fujitsu Ltd. are constructing an 18,000 mile undersea WDM optical fiber cable looping from Australia and New Zealand to the U.S., then back through Hawaii and Fiji to Australia [Ref. 6].

B. HOME NETWORKS OF THE FUTURE AND THE HOME COMMUNICATOR

Thus far the discussion has centered on the technology presently available that has improved the operability of the worldwide internetworking business over the past few decades. Within the information technology field it has not been standards such as IEEE that have shaped technology evolutions but instead consumer practices that have dictated which standards take hold of the business world environment and flourish. As technology increases a general trend has seen business-type networking environments being introduced into the home environment at an increasing rate. Individual homeowners are realizing benefits that only the business sector has seen in the past. Networking solutions within the household are now able to utilize one ISP connection and span it out to several computers for simultaneous schoolwork, business work, and/or entertainment purposes.

A short time ago in order for the homeowner to set up and employ a networked environment within his or her home it has been imperative that he or she has extensive experience and knowledge of networking. Networking is not intuitive and the different interactions between Interior Gateway Protocols (IGP) and Exterior Gateway Protocols (EGP) can be confusing to an untrained individual. The business world has begun meeting this challenge by creating hardware and software networked systems that are easier to operate regardless of the level of experience that the consumer possesses.

Optical fiber has been a backbone medium utilized by telephone companies for over a decade now because of its exceptional bandwidth capabilities. Today optical fiber is no more expensive than twisted pair cable. Yet, several inhibiting factors have prevented the use of optical fiber within the average household. Today a majority of the industrialized nations have comprehensive telephone networks. 90 to 100% of the households within these industrialized nations already possess telephones with an existing twisted pair infrastructure up to and within the household. The cost to rewire households with optical fiber has thus far outweighed any added bandwidth benefits to the consumer. Instead it has been more convenient to utilize the existing telephone wire and establish a new cable infrastructure to include television cable service to the household. In addition, existing telephone and television service within the industrialized nations has proven fairly inexpensive for the average consumer.

As computer and internetworking technology continues to progress bandwidth requirements for the average user in the industrialized countries will increase as well. Data transfer over internet has already been integrated with voice over internet and video teleconferencing. Each of these services within the home will add increasing strains on the existing household networking medium due to the time sensitive necessities along with increased bandwidth requirements. As the computer trend continues to grow the natural next progression is to network the household communication system with the entertainment system. Instead of receiving television and telephone service separately the two will be combined. The existing household twisted pair will not be able to maintain such a networked household environment. Optical fiber rewire of households will be considered as the viable solution to accommodate the future fully networked household.

Another scenario that must be noted is the situation within developing countries. The existing telephone and television services can be drastically more expensive due to monopolies caused by political corruption. In this atmosphere usually there is little existing infrastructure to the majority of individual households for telephone or television service because of the substantial expense to the middle and low class clientele. With this type of an environment together with the history of growth in computer technology it would be more prudent to design a complete networked infrastructure utilizing optical fiber technology.

Finally, in order to take advantage of the future networked household there must be a device that performs as the networking core. Such a device would be called the home communicator. It will deliver the previously discussed innovative networking improvements down to the customer in the form of consolidated functionality. Presently customers are required to get service from the phone company in order to receive telephone service and cable service from a television cable company in order to receive an increased quality of television reception. In addition, utilization of a cable or telephone ISP would give the customer e-mail and internet capability with either the television or telephone, but no service presently offers all three in one. The home communicator would allow the two presently unrelated functions of television and telephone service to be contained within one package, one service with the added internetworking capability. Figure 1-1 demonstrates the functionality of the home communicator.

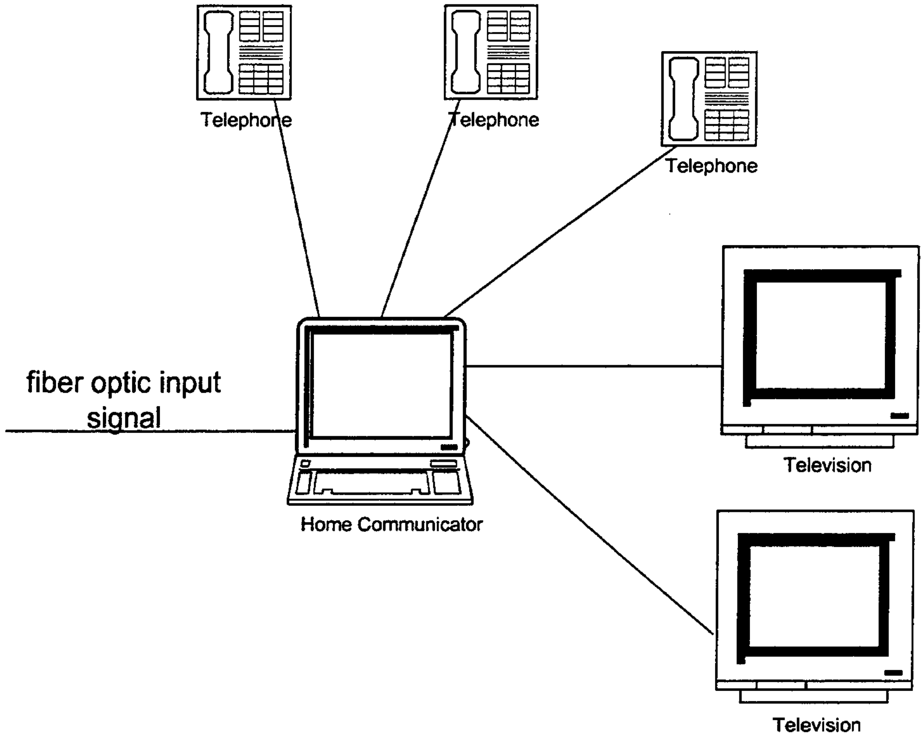


Figure 1-1 Home Communicator Functionality

The optical fiber household network in conjunction with the multipurpose multifunctional communicator would allow one input into the house with the ability to have up to several distinct televisions and several individual telephone lines as well as providing e-mail and web browsing capabilities to the customer. It will be designed as an easy to set up and use instrument. The customer is not required to have networking experience in order to operate the home communicator. Its functionality will be the most basic for the original lowest costing package. Functionality will be increased with greater priced packages depending on the affluence of the household. More televisions and telephones can be added to create the networked environment with added functionality to the basic model. Each telephone can be used independently or in unison. Figure 1-2 illustrates the household with a future network utilizing a home communicator design.

C. DOCUMENT DESCRIPTION

Chapter II of this thesis describes the computer operating system Linux and other software packages that will increase capability to the home communicator. Chapter III discusses Internet Protocol version 4 and the enhancements of the protocol in the area of individual packet transfer for IP version 6. Chapter III also investigates the IPv6 addressing scheme for a possible regionalization proposal. Chapter IV introduces the high-level requirements and high-level design for the home communicator. Chapter V describes a six-month plan of execution to produce the home communicator software and hardware package. Chapter VI is the conclusion.

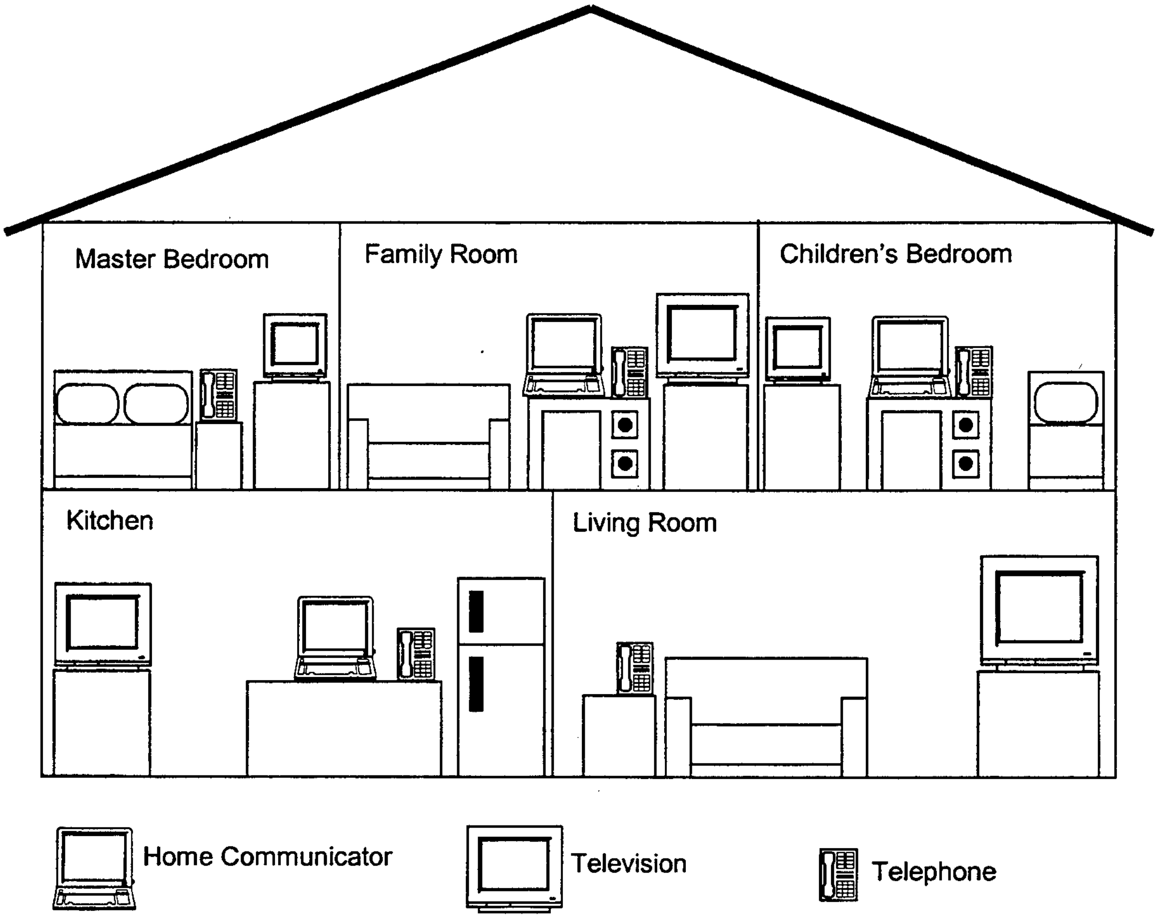


Figure 1-2 Future Household with Home Communicator and Optical Fiber Network

THIS PAGE INTENTIONALLY LEFT BLANK

II. LINUX OPERATING SYSTEM

A. INTRODUCTION TO LINUX

Linus Torvalds developed Linux based on the operating system UNIX. Linux is available under the GNU General Public License. The first version of Linux, 0.02, was available on the internet in October 1991 [Ref 7:p. 6]. Linux releases are controlled differently than any other popular commercial operating system. Inputs from computer programmers all over the world are received, tested and added to the kernel for next release if found acceptable. Linus Torvalds controls which contributions become part of Linux releases.

The new trend of operating system programming design has moved towards microkernel technology. Microkernel technology utilizes object-oriented modules to perform the system functionality. The actual kernel module provides the necessary minimum functionality, inter-process communication and memory management. Remaining functions of the operating system are relocated to autonomous processes. These processes communicate with the kernel through clearly defined interfaces. The drawback to this philosophy is that communication between these secondary functional modules is comparatively time intensive. This design requires faster hardware to prevent noticeable delays in processing. Microsoft is following the microkernel design for its present and future Windows packages. [Ref 8:pp. 16-17]

In the past, design architecture for operating systems has focused on monolithic kernel architecture. Time optimization is at the heart of the monolithic kernel. The kernel itself has all the functionality of the operating system within its coded module. Communication between the different function elements within the kernel is a simple function call. Although Linux' kernel is modular in design the overall philosophy of Linux design utilizes this classic monolithic architectural design. Many functions of the kernel that could be removed for increased modularity remain to within the Linux kernel because it reduces the process time. The monolithic design affords Linux the ability to function well on a lower end computer such as an Intel 386 as well as high end Pentium processors. [Ref 8:pp. 16-17]

1. Comparison of Linux to Microsoft and Apple Computer

Several characteristics of presently available operating systems are compared here that will support requirements of the home communicator and defend a conclusion of Linux' superiority for this particular purpose. These requirements include low cost, software storage and operation size, stability, and ability to configure the operating system to sustain IPv6.

Linux is less expensive than Microsoft or Apple Computer products. Microsoft requires a licensing price for all of their software and Apple Computer hardware and software packages start at $1,599 [Ref 9]. Linux utilizes a GNU General Public License. The only costs for opting with Linux would be developmental. Also, Linux requires much less hard drive memory and RAM to run than Microsoft. The minimum requirement for the WINDOWS 2000 server solution hardware is 1 Gig hard drive and 256 MB RAM [Ref 10]. The minimum setup of the Power Mac G4 is 10 Gig hard drive and 64 MB RAM [Ref 9]. Linux can easily be run on a 500 Meg hard drive with 16 MB RAM due to its monolithic background design.

It is no secret that Microsoft Operating systems have a poor reputation for long-term user stability. Linux on the other hand has a strong reputation for stability. Apple Computer has also enjoyed a very solid reputation for its equipment and software. However, a detracting factor is that Apple Computer requires the purchase of their hardware and operating system software together.

A final aspect of Linux that outweighs its counterparts is the ability for anyone to configure IPv6 into the Linux operating system. Microsoft and Apple Computer have not offered the IPv6 functionality up to this point. Microsoft has only just recently offered an IPv6 beta version stack [Ref 11]. Linux on the other hand, is openly configurable due to the GNU General Public License.

2. Linux Distributions

Linux can be acquired through distributions. There are several major distributions of Linux available today. Any company can create their own distribution as long as the source code for the Linux kernel portion is made publicly available. Often a company will create a Graphical User Interface (GUI) for installation ease and/or other small software programs to run with Linux as part of their distribution package. In this case it will cost more than a nominal DC-ROM creation fee to the customer. This GUI for installation and/or the additional software is usually not included in the GNU General Public License and is considered proprietary. In this case a license would have to be purchased for computers the software is loaded on.

For the purposes of this study there is no requirement for ease of customer installation. The goal here is to create a software system and image the file to install on all other manufactured systems. The customer would not do the installation, the hardware system would come to him or her preloaded with software and ready to plug in and turn on.

Overall there are approximately 25 fully capable English distributions of Linux available today. A few of the major distributions are enhanced and released by other companies. For instance, Corel, Libranet and StormLinux 2000 all utilize the Debian release for their basic Linux kernel. Some of the most popular versions include Debian, Red Hat, and Slackware. Each of these adds their own additional software programs and/or their own GUI installation packages as a supplement to the Linux operating system. The majority of the Linux distributions cater to IBM clone processors although the MkLinux and PowerPC 2000 are strictly designed to work on the power Macintosh platforms. Presently on the same web page there are 14 mini and specialty distributions. Several of them are specifically created to operate as a secondary operating system to the Windows environment. [Ref 12]

3. Investigation of Existing Linux Distributions

It is possible to download a complete version of the Linux operating system from any of the various Linux distribution web pages. The documentation to load and configure the software is also available through web pages such as: http://MetaLab.unc.edu/LDP/HOWTO. The earlier versions of the Linux kernel are not supported as well as the latest 2.0 versions. Therefore it would be prudent to utilize the latest 2.1 Version to develop the software package. A hardware system could be designed easily and cheaply in today's environment to accommodate the RAM and memory storage requirements of the newer and larger Linux kernel.

In order to test feasibility of obtaining Linux without cost the author downloaded Linux from web sites listed in Table 2-1. Debian CDs were constructed from a Debian CD-ROM setup guide then loaded onto an Intel Pentium 100MHz computer. The installation of a basic kernel was accomplished by following the menu driven selection process. Each distribution investigated had its own GUI installation software but all installation GUI's accomplished the same functions. The next section discusses the various GUI interfaces available for the Linux kernel.

Company Name	Home Web Site
Debian:	http://www.debian.org/
Corel:	http://linux.corel.com/
PowerPC 2000:	http://www.linuxppc.com/
White Dwarf:	http://www.emjembedded.com/

Table 2-1 List of Linux software Home Web Sites

4. Linux Desktop Environment Examination

Linux has been well known for its utilization of a command line interface. There has not been a substantial Linux GUI desktop solution available to the customer until recently. Where Microsoft and Apple Computer have developed operating systems with the window manager and desktop environment built in, Linux has remained a command-line based environment. Other software packages have been developed that offer a graphical user interface functionality but these software packages actually operate separately from the UNIX and Linux kernels.

Several of the more popular GUI software packages for the Linux kernel are listed in Table 2-2. The software GUI systems listed to the left side of Table 2-1 must be installed and functional prior to the software systems listed to the right of the chart. In other words any one of the Window Manager software operates as an add-on to the X-Window System; the Desktop Environment software requires both an operational X-Window System and a Window Manager to be installed beforehand. Each of these software packages described is included under the GNU General Public License and can be downloaded from parent company sites for installation on Linux systems.

The X Window System was the first attempt at a GUI for UNIX systems. It took shape in the late 1980's prior to Linux' inception. X Window System is not technically a desktop environment or a window manager. Instead, X Window System was developed for the UNIX environment with the goal of allowing applications to be running from the one computer yet be displayed on more than one computer at the same time [Ref 7:p.

162]. Although it provides a windowing environment for Linux it alone is lacking certain functionality that is required in day-to-day performance of a desktop environment or window manager.

	X-Window System	**Window Manager**	**Desktop Environment**
Provides:	basic window evolutions only	opening, resizing, and closing "function" or "program" windows	overall user control including drag & drop, paneling, file management, etc.
Specific Name(s):	XFREE86	twm fvwm Enlightenment *	GNOME KDE

* to maintain brevity other window managers readily available are not specifically discussed here

Table 2-2 Linux GUI and windowing software

X Window System possesses several deficiencies for a normal computer user. X Window System does not support the ability to drag icons onto the desktop, does not have either an integrated file manager, a unified help system for applications, or utilities such as clocks, calendars, and calculators. It was not developed to offer desktop functionality and therefore lacks the actual capabilities that would be required by a standard computer workstation consumer. XFree86 is the edition of the X Window System that supports the Linux platform. Note that the protocol X Window System utilizes between the server and client computers is TCP/IP even when directly connected [Ref 7:p. 15].

In order for X Window System to ever approach the functionality and feel of Windows or Macintosh it is necessary to utilize a window manager in addition to the X Window System. The window manager would embody the layer of software between the customer and the X Window System software. The X Window System software would correspond to the interface between the window manager and the Linux kernel. The X Window System software is required to run the window managers available for Linux.

Overall, the window manager for Linux provides features for opening, resizing, and closing function or program windows. All of the window manager packages allow the user the functionality to configure the appearance of the window frames to his or her specific requirements. Window managers offer such provisions as titlebars, shaped windows, user defined macro functions, icon management, and user configured key and pointer button bindings. There are many window managers available today and most come under the GNU General Public License. Some of the most popular include twm (Tab Window Manager), fvwm (Virtual Window Manager), and Enlightenment, which are described here.

twm is the only window manager that is distributed with the X-System or XFree86 package. twm offers the window manager functionality discussed above. When started without any other type of session manager twm will execute to the foreground as the last client. When twm is terminated the session is automatically terminated. twm is more difficult to handle than other window managers that have been developed for Linux. [Ref 13:p. 263]

fvwm was derived from twm. fvwm is a popular window environment with Linux because it combines ease of use with low memory consumption. fvwm utilizes approximately one half to one third the memory consumption of the twm window manager. fvwm is available in three versions, 1.2n, 2.0, and fvwm95. Version 1.2n has the lowest memory consumption. Version 2.0 offers more configurations possibilities. fvwm95 has a similar environment impression to existing Windows and Macintosh operating systems. [Ref 13:p. 272] Several other window managers not described here are derivatives of fvwm such as AfterStep and LessTif [Ref 14:pp. 68-69].

The Enlightenment Window Manager supports all the functionality described above. The software package is not a derivative of twm or fvwm. Compared to other window managers Enlightenment offers added user configuration of the function and program windows by offering customization themes to the user. These themes allow the user to create individual window environments unequaled by other window managers. Enlightenment is presently the only window manager 100% compatible with the desktop environment GNOME described below. [Ref 14:pp. 69-70]

Desktop Environments are the next step in the GUI creation for the UNIX and Linux operating systems. Where window managers control the function and program windows for the X Window System the desktop environment offers overall user control down to the underlying Linux operating kernel. The desktop environment utilizes a specific or several window managers to support the window functionality within the desktop view. The desktop environments offer the following features to X-Windows functionality: Drag and Drop, Drive Icons, Paneling, Integrated File Management, Integrated Session Management, and Inter-process Communication between Applications [Ref 14:pp. 74-77].

The desktop environment adds capabilities to the window manager's functionality. One added capability by the desktop environments not found with window managers alone is the ability to start often used programs from a menu. Another desktop environment benefit is the distinct applications developed for desktop usage such as calculator, simple text editor, and file manager. Once the Linux, X Window System, Window Manager, and Desktop Environment are set up it appears as one intact package to the customer. The most common X-Windows window environments available are GNOME (GNU Object Model Environment) and KDE (K Desktop Environment). The following are short descriptions of these Linux desktop environments.

GNOME stands for GNU Object Model Environment. It was developed from the beginning on open-source principles. GNOME utilizes the GNU toolkit called GTK+ for development. GNOME does not have its own window manager but can be utilized with any window manager that is GNOME compliant. Presently only the window manager Enlightenment is 100% GNOME compliant. [Ref 14:pp. 74-77]

GNOME's look and feel is similar to the Windows or Macintosh desktop environment. It allows items to be placed on the desktop that can be manipulated using right click menus or execution by a left click of the mouse. GNOME is not quite as finished a product as either of the other two major operating system companies, but the same functionality found in Windows or Macintosh environments is present within the GNOME GUI.

GNOME has a menu panel across the bottom of the screen that can be programmed to hold buttons for most often used operations. This menu panel is in addition to the programmable desktop icons found on the desktop portion of the screen. The menu panel would be particularly useful for the customer utilizing the home communicator. The customer could have full view of other available applications while another deployed application might have covered the desktop icons. A standard menu panel could be developed to deploy all the functions from individual buttons across the bottom menu panel. [Ref 15] GNOME desktop default offers 4 separate desktops simultaneously and can be configurable up to 64 desktops. The bottom panel possesses the desktop selection screen for the customer's convenience.

The GNOME desktop is highly configurable. The customer can either utilize the design provided or they can reconfigure the buttons and icons to support personal requirements depending on their level of computer knowledge. GNOME also contains a set of standard desktop tools and applications including a simple text editor (gEdit) and chat (xchat IRC client). These specifically support requirements for the home communicator concept. [Ref 15]

KDE is short for K Desktop Environment. KDE is usually operated with the window manager kwm. KDE utilizes the Qt toolkit for development. Initially, Qt toolkit did not fall under the GNU General Public License, but the license has since been modified to fall under GNU. KDE desktop comes under the GNU General Public License. KDE is more mature than GNOME specifically with the KDE applications. KDE was begun as a project in 1996 by German LyX developer Matthias Ettrich. The developer base quickly grew into a group of interested programmers similar to the growth of Linux itself. The first stable release, 1.0, was released in July of 1998 and KDE is presently on release 1.1. [Ref 7:p. 612]

Similar to GNOME, KDE has a look and feel that approaches Windows and Macintosh operating systems. KDE also allows items to be placed on the desktop that can be manipulated using right click menus or execution by a left click of the mouse. KDE displays buttons on the lower panel that correspond to available applications. Like with GNOME the KDE panel can be configured to display the applications available to the customer on this bottom panel. If the desktop is covered with a deployed application the bottom panel allows the customer continuous view of his or her options.

There are some additional qualities found in KDE that are not found with GNOME. KDE has the same bottom panel that allows activation of common programs and the capability of configuring up to 64 simultaneous desktops. Yet, KDE has a second panel across the top of the screen that holds an individual button for all activated applications. When a button on the top panel is pressed by a mouse click the screen automatically moves to the desktop the button's program is located on. GNOME requires the customer to remember which screen the program is located on and specifically request the screen to get to the application.

Similar to GNOME, KDE is highly configurable. The customer can either utilized the design provided or they can reconfigure the buttons and icons to support personal requirements depending on their level of computer knowledge. KDE also contains a set of standard desktop tools and applications including a simple text editor (Text Edit) and chat (Chat Client – ksirc). These specifically support requirements for the home communicator concept. [Ref 16]

B. APPLICATION SOFTWARE FOR LINUX

Although Linux is not as popular as Microsoft or Apple Computer there are a growing number of companies that are developing applications compatible with the Linux operating system. There is a requirement to locate functionality for web browsing, IP telephony, e-mail, chat, and text editing. In order to develop the home communicator software package it is necessary to canvas existing application software available to the Linux system.

1. Web Browser

Netscape and Internet Explorer are the most popular browsers available today. Internet Explorer, does not offer a version that supports Linux. Internet Explorer is available for HP-UX and Solaris only and cannot be used with Linux without major modification [Ref 17]. Therefore, Netscape is investigated here as a possible application to support the home communicator.

Netscape presently only has an English version available for the Linux operating system [Ref 18]. This would affect the usage of the home communicator in foreign countries. The software will function, but the view of the Netscape browser itself will be in English. For the countries that use oriental languages there is a greater concern with Netscape. The oriental alphabet requires two bytes of code for each character where English and European languages only require one byte of code for each character [Ref 18]. The English version of Netscape cannot be utilized with the oriental version of Linux for this reason. They are incompatible. Until Netscape develops an oriental language version Netscape could not be utilized for oriental versions of the home communicator.

Netscape has other functionality to offer to the home communicator. There is a chat and e-mail function within the full Netscape Communicator package. Netscape functionality for these requirements will be discussed below within each individual section.

2. IP Telephony

There are several peripheral items to consider when attempting development of a Voice over IP (VoIP) system. The first major decision for VoIP is the designed footprint of the customer base. If the VoIP application is functional only home communicator to home communicator such as the chat service is computer to computer then a problem arises that persons purchasing the home communicator can only converse with others that already own the home communicator; this is definitely unsatisfactory from a customer standpoint. In order to effectively retail this technology to the consumer it would be important to literally flood a customer base with the product. Within industrialized nations this would be a difficult task. The existing public telephone service is so prevalent that it would be difficult to sell the home communicator. Yet, in a country that is deprived of low costing telephone service due to existing monopolies the home communicator could possibly be flooded into the market if its cost was reasonable.

Speak Freely offers this type of VoIP technology. Speak Freely is offered as Open Source Software, with no purchase required. It allows computer users to connect to other computers presently on the internet. This technology does not work in conjunction with the public telephone system, though. It presently supports only on line computer to computer connection. The Speak Freely customers utilize a Look Who's Listening server to publish their information into searching directories. Another point worth making here is that presently the UNIX version of Speak Freely is not GUI supported. The present UNIX version is 7.0 and it is a command line application. There is a working group presently developing an X Windows upgrade to the UNIX package of Speak Freely. Other companies such as PhoneFree supply similar software for no cost to customers. However, they do not support the Linux operating systems. [Ref 19]

On the other hand, if the system is going to be interactive with the normal public phone service there must be an interface point to the public phone system. This requires a company offering the VoIP service to develop and deploy a gateway infrastructure supporting the link to the public telephone system or to utilize just such a low cost service from a preexisting service company in order to offer VoIP to the customer at a low cost. The cost of utilizing such a service from another company would impact home communicator service costs. This would offer a larger VoIP footprint to the customer than Speak Freely offers.

Dialpad.com displays this type of VoIP technology that is available today. Dialpad.com is a web site that offers free telephone calls anywhere in the United States through the computer. Anyone would utilize the web browser, such as Netscape to access the dialpad.com web site. Once on the dialpad.com site it is a simple matter of requesting a free account by following a simple registration process. The only requirement for the user is that the computer they utilize has a speaker and microphone. One drawback to this technology is that persons can only call regular public service telephones. [Ref 20] For instance, there is no capability with dialpad.com to call another computer set up with dialpad.com access. Presently dialpad.com is only available for use within the United States.

Another technology presently available for this type service is IP telephones. Cisco has released IP telephones that presently cost approximately $400. These telephones look and perform as a normal telephone, but they plug directly into an existing LAN with an IP address. They possess a regular telephone number. These telephones require an entire infrastructure to be connected to the public telephone system. This overall required computer server infrastructure is quite costly. With IP telephones within the home communicator the functionality to call home communicator to home communicator, home communicator to pubic telephone customer, and public telephone customer to home communicator is realized, though. [Ref 21]

There is an option that would require investigation. This option would support the functionality found with IP telephony utilizing standard telephones. The home communicator would be configured as a server and each telephone connection within the home communicator would utilize individual NIC cards. Each NIC card would possess an individual IP address. The server itself (home communicator) would process the incoming telephone signals and pass onto the telephone only the raw sound IP packets. Each NIC IP address would correspond to individual telephone numbers. There would be a different number for each telephone and another number for all the phones if the caller wanted to locate anyone within the house instead of a specific person within the household. This type system along with IP telephones offers the largest footprint to the home communicator customer. This technology is not presently available, though, and would require extensive design and development.

3. E-mail Capability

E-mail capability requires support to the customer by the ISP. The DNS server name (for example, "@home.com") for each customer must be obtained from the ISP as a portion of the monthly service fee. In this case the home communicator would be sold as the hardware to be used with the home communicator ISP service to support the 24 hour 7 day per week internet connection. If the home communicator is set up like a mail client the customers would require mail storage on mail servers at the ISP location even though the home communicator is designed to be active 24/7. If the home communicator were configured as both a mail server with the mail client the customer would not be required to maintain the client open 24/7. The mail server would store the incoming e-mail whether or not the mail client was activated. If the home communicator is configured with the mail server concept the ISP would require only mail transfer agents instead of mail server assistance.

The actual mail client software can be found within several of the software packages discussed above. E-mail client capability is offered with Netscape Communicator. Netscape supports POP3 and IMAP protocol mail accounts. Netscape presently supports an offer of a free mail account through USA.NET to anyone who registers. The free mail account only permits up to 5 MB free mail storage and considers an account inactive after 90 days of no activity. There is no guarantee as to how long other free services such as the USA.NET will be offered and thus cannot be trusted as a viable option for clients. [Ref 18] KDE desktop also has an e-mail capability application called kmail. kmail will support the same functionality of Netscape mail client service except it does not support IMAP accounts.

In order to maintain consistency of the home communicator software package the KDE desktop e-mail client application (kmail) should be utilized for the home communicator regardless of how the mail server concept is managed. It would be necessary to test feasibility of mail server, Linux server, and mail client workstation all within the same Linux operating system prior to actual deployment of the concept. Limitations should be set on storage size with automatic timed deletion of e-mails on the mail server.

4. Chat

Chat allows persons utilizing the internet to contact others that are presently on line and registered with a chat server. Once a connection between customers is made Chat allows the users to converse by typing information and pressing the Enter key to send. Chat functionality can be added to the home communicator through any of several avenues. Netscape supports chat functionality directly from the top menu as Instant Messenger. Anyone can sign up for free user status. Netscape chat functionality does not require the web browser to be opened in order to run Instant Messenger. GNOME and KDE desktop environments also support chat functionality through their own applications, xchat IRC client or Chat Client respectively. In order to simplify the home communicator for the non-savvy computer users a menu panel button would be programmed to deploy the chat function regardless of which version, KDE or Netscape is utilized.

5. Simple Text Editor

Simple text editor capability can be added directly through the desktop environments of Linux. There is a simple text editor function included within both KDE (Text Editor) and GNOME (gEdit) software packages. During development of the home communicator package the application for simple text editor would be included. In order to simplify the home communicator for the non-savvy computer users a menu panel button would be programmed to deploy the simple text editor.

Another possible choice for simple text editor capability that should be noted here is Star Office. Star Office is another GNU General Public License software compatible with the Linux operating system. Star Office is a Sun Microsystems product that offers the common desktop functions of word processor, spreadsheet, and slide presentation developer. This product would far surpass the simple text editor requirement but would greatly increase a user's capability with no added cost. It must be noted, though, that it would also increase the complexity of product operation as well as force an increase in the hardware requirement for hard drive space and Ready Access Memory (RAM).

THIS PAGE INTENTIONALLY LEFT BLANK

III. INTERNET PROTOCOL, VERSION 4 VERSUS VERSION 6

A. IPv4 BACKGROUND

The first packet switch network was a four-node packet switching network known as Advanced Research Projects Agency Network (ARPANET). It went into operation in 1969. The protocols utilized for the original ARPANET were subject to network crashes and operated rather slowly. In 1974 a new suite of internet protocols known as TCP/IP was proposed by Vinton G. Cerf and Robert E. Kahn. By 1983 all existing ARPANET nodes (approximately 300) were converted to these new TCP/IP protocols. In 1983 the Department of Defense adopted TCP/IP as its protocol standard, which had spawned a large market for the technology. Throughout the 1980s the government funded various UNIX developers to build the TCP/IP suite for UNIX systems. Regional Service Provider organizations developed around the world to support connections to the internet. The ARPANET remained the backbone of a growing evolution of academic and commercial research networks. By 1994, when the Internet was officially changed from a research testbed to a commercial service network it was made up of millions of interconnected computers. Today IPv4 is utilized throughout the internet. [Ref 22:pp. 10-11]

The IP protocol utilizes encapsulation to deliver data. A simplistic definition of IP protocol follows. IP operates by accepting data from the next higher protocol, either TCP or UDP, creating a datagram, routing it through the network, and delivering it to the recipient host and then pertinent application. IP uses the subnet mask and IP routing tables to deliver the datagram to the next router or host on the path to the destination. The subnet mask helps determine whether or not the source node is on the same LAN as the destination. The routing table designates how the IP packet is routed when the destination node is not on the same LAN as the source node. All routers and hosts connected to a network and the internet have a routing table which defines the nodes within range of it. By routing to the next hop designated within each router or host routing table the datagram is transferred from source to destination.

Figure 3-1 illustrates the fields required for an IPv4 datagram packet. The most important fields within the IPv4 header are version, source IP address, destination IP address, and protocol. The version field establishes the version of IP being utilized. The source IP address and destination IP address designate where the datagram is coming from and where it is going to by IP address. The protocol field transfers the data packets to the appropriate application service within the destination host such as TCP or UDP. Other fields such as the Precedence/Type of Service and the Option fields offer additional options to IP. Three bits hold precedence levels are 0 through 7, 7 being the highest priority. Four bits hold type of service to depict options such as: minimize monetary cost, maximize reliability, maximize throughput, minimize delay, or maximize security.

Other options available with the Options field include possible routing controls, basic security, extended security, and router alert. Routing controls include strict source route, loose source route, and record route. Strict source route as it sounds describes a complete path that must be followed to the destination while loose source route designates milestones along the way. Record route contains the list of IP addresses of routers that were visited by the datagram. Basic security assures that the sending host is authorized to transmit it, intermediate routers appropriately relay it, and that receiving hosts are allowed to receive it. The option parameters for basic security range from Unclassified to Top Secret. There are flags that identify the protection authority, such as Central Intelligence Agency or Department of Energy, whose rules apply to the datagram. The extended security field can be found with the basic security option field.

There are several different formats for this option, depending on the needs of the defining authority. The router alert is to let routers along the way know that they should examine the datagram carefully because of special processing requirements.

```
0                 1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver |   HL  |Type of Service|          Datagram Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Datagram ID              |Flags|      Flag Offset          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    TTL       |  Protocol     |           Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source IP Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           IP Options (with padding if necessary)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                   Data Portion of Datagram                    .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                          (Payload)                            .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-1 IPv4 Datagram Packet Header Format

Other fields within the IPv4 header that support fragmentation of IPv4 packets include header length, length of datagram, identification, flags, and fragment offset. The header length is necessary since use of the option fields will create a longer header when utilized. The IPv4 header can be 20 bytes to 60 bytes. The length of datagram is important for network traffic. If during the routing of the datagram packet a network maximum frame size prevents the pass through of the existing packet it must be fragmented. The identification field consists of a 16-bit number that designates fragmented datagram packets as initially coming from the same packet.

Flags and fragment offset fields work together to describe the specific location of the fragment within the original packet. One bit within the flag field indicates whether or not the packet may be fragmented. The fragment offset is a 13-bit field that displays what is known as the fragment block. The original datagram packet is broken into 8-bit blocks known as fragment blocks. The 13-bit field displays this fragment block value from 0 to 8191, which directly corresponds to a 0 to 65,528-bit offset from the original datagram packet. Another bit within the flag field shows if this is the last fragment of the original datagram or more come after this particular block.

Finally, the time to live and header checksum fields within the IPv4 header are utilized for network cleaning functions. The time to live limits the amount of time any datagram packet is alive on a network. The time to live value will decrement at each router. If it reaches zero before it reaches its destination it is discarded. The header checksum field is a checksum value of the header. It must be recalculated at every router due to the time to live function and when the packets must be fragmented.

With the options described above IP version 4 has proven very reliable for data transfer up through the 1980s and 1990s yet IPv4 has become less efficient as the internet has continued to expand into the new century. IPv4 was not designed to manage the extent of nodes it supports today. Up through the past few years use of the VoIP convention has greatly increased. As the personal computer market has expanded in the past decade so has the requirement for a larger internetworking solution for TCP/IP. Over the past 5 years there has been efforts to create a next generation TCP/IP to incorporate requirements of today's internet. IPv6 offers many more distinctive attributes, specifically in the area of voice/video over the internet.

B. IPV6 ENHANCEMENTS FOR PACKET TRANSFER WITH VoIP

The actual operation process of IPv6 is the same as IPv4. In order to phase IPv6 into operation IPv6 can be utilized within an IPv4 environment yet with a degeneration of full IPv6 functionality. IP with version 6 will sustain voice and video over the internet in several ways that version 4 does not. In order to anticipate an expanding internet IPv6 has included simplification of the IP header to realize increased operability from the internet. The IPv6 simplified header allows more functionality with an overall smaller overhead requirement for processing. The new header fields available with IPv6 assist by prioritizing packets and associating packets within a flow for site-to-site message traffic. Finally, the difference in fragmentation handling between IPv4 and IPv6 also assists with voice and video IP messaging.

1. Simplified Header

IPv4 utilizes standard header fields that offer the consistent options described above. All routers and hosts along the path as well as the destination host still must process all the fields within the IPv4 header whether or not they are utilized, such as with the fragment offset even if the packet is not a fragment. IPv6 utilizes a slightly different model for its datagram header and IP processing. The IPv6 header has fewer fields than the IPv4 header. Figure 3-2 depicts the fields required for an IPv6 datagram packet. Another important difference is that the IPv6 header is a standard size. This precludes the requirement for a header length field in the IPv6 header as with the IPv4 header.

IPv6 utilizes "extension" headers for each datagram packet that requires special options. These extension headers are not actually part of the IPv6 header. The "Next Header" field delineates the next header for each IPv6 datagram. The extension header resides within the datagram packet before payload portion of the datagram and is not part of the header itself. Figure 3-3 illustrates the extension header concept and Figure 3-4 shows the standard extension header format. The IPv6 specification (RFC 2460) recommends that the next headers be placed in a specific order. The required order is IPv6 header, hop-by-hop extension header, destination options header, routing header, fragment header, authentication header, encapsulation security payload header, destination options header, and finally upper layer header. Thus, when there are no additional IP options required the next header designated would be the higher protocol such as TCP or UDP. The destination options headers and routing header deal with source routing, which are IPv6 options outside the scope of this thesis and not discussed further.

```
0                 1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver | Traffic Class |              Flow Label                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Payload Length         | Next Header  |   Hop Limit     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Source IP Address                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                  Destination IP Address                       |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                  Data Portion of Datagram                     .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                          (Payload)                            .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-2 IPv6 Datagram Packet Header Format

IPv6 has two categories of extension headers. One category requires processing by every node between the source and destination while the other category requires processing by the destination host only. Figure 3-3 shows the difference between these two type headers by showing node perception along the IPv6 datagram path. The "hop by hop" extension headers are the only extension headers that are processed by each intermediate node as the packet transverses the internet. The hop-by-hop extension header is the first next header that follows directly after the IPv6 header. At this time the only two options specified for hop-by-hop are Jumbo payload option and the router alert option. [Ref 23:p. 83] Neither of these existing options improves VoIP or video teleconferencing quality.

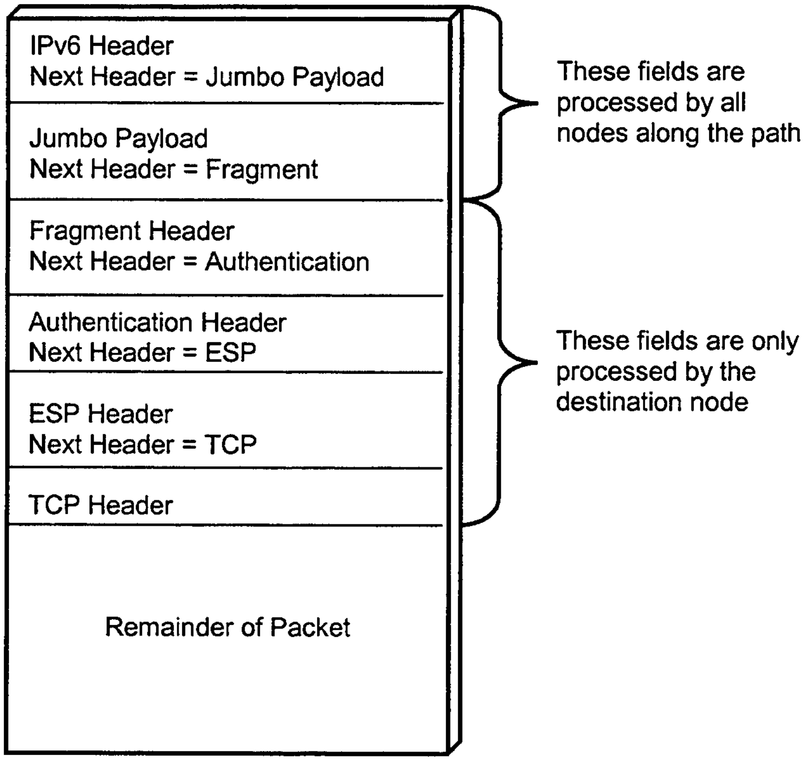

IPv6 Header
Next Header = Jumbo Payload
Jumbo Payload
Next Header = Fragment
Fragment Header
Next Header = Authentication
Authentication Header
Next Header = ESP
ESP Header
Next Header = TCP
TCP Header
Remainder of Packet
These fields are processed by all nodes along the path
These fields are only processed by the destination node

Figure 3-3 Extension Header concept holding 4 extension headers

```
0                   1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header |  Hdr Ext Len  |                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                 +
|                                                               |
.                           Options                             .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-4 Standard hop-by-hop / destination Extension Header Format

No other extension headers, including higher level protocols, are processed by intermediate nodes. Instead the remaining extension headers are processed only at the destination node. At the destination node all extension headers are processed one by one. After all of the extension headers have been processed for an individual packet the next header field will show the next header protocol, such as TCP or UDP required for processing the packet.

2. Fragmentation

The concept of datagram packet fragmentation for IPv6 is managed fundamentally different than for IPv4. IPv4 packets can be fragmented at any intermediate node along the path that does not allow packets as large as the original packet size. IPv6 allows fragmentation only at the originating or source node. The fragmentation option is utilized as an extension header with its own format.

Figure 3-5 portrays the fragmentation header format. The next header field (8 bits) is the format of the subsequent header field. The next field (8 bits) of the fragmentation header is reserved for future use. The fragment offset (13 bits) value tells the destination numbered in 8-bit segments where this portion fits within the fragmentable portion of the packets.. The fragmentable portion includes only the payload and extension headers that are to be processed when the packet has arrived at its final destination. The next field is another field reserved (2 bits) for future use. The next field is the M flag which when the value is 1 indicates another fragment is forthcoming, a zero indicates this is the last fragment. The final identification field holds a 32-bit identifier that is intended to uniquely identify any packet sent recently.

As described above this principle simplifies the header while it greatly reduces overhead of router processing. Since fragmentation is not a hop-by-hop extension header only the source and destination node ever process the fragmentation information located within the payload of the packet. Figure 3-3 portrays this. It must be noted that since datagram packets can not be fragmented along the way when a node cannot forward the packet due to packet size limitation an error message must be sent back utilizing ICMPv6.

```
0                   1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header |    Reserved     |    Fragment Offset      |Res|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Identification                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-5 Fragmentation Header Format

In order to prevent error messages the IPv6 specification recommends all nodes implement path MTU discovery and send packets within the size allowed per the path utilized [Ref 23:p. 124]. Simply put, the MTU discovery is a process where the originating node sends a packet to the destination. Along the path it checks the largest allowable MTU size of each node and delivers this information back to the originating node prior to releasing any datagram packets to the destination. The packets now sent from source to destination will be no larger than the intermediate nodes with the lowest capability of MTU size discovered through the MTU discovery process. As described above ICMPv6 will be utilized if instantaneous changes occur within the network path to preclude usage of the MTU discovery path.

3. Added Functionality within the Header Fields

The IPv6 header contains several fields that are not found within the IPv4 header. The functionality of these additional fields expands the usefulness found within IPv4 headers. These added capabilities directly support the extra requirements thrust upon the internet due to VoIP. The traffic class and flow label fields within the IPv6 header prioritize packets and control delivery paths in an attempt to improve upon IPv4 technology. Figure 3-2 illustrates this format of the IPv6 header.

The traffic class field is an 8-bit value. The traffic class field replaces the type of service field found in IPv4 with added functionality. Presently the traffic class is marked to be a priority field in which one of 16 different priority classes could be specified for each datagram packet. The traffic class is identified by the originating source and can be changed by intermediate routers that support the traffic class functionality. All other routers would merely pass on the existing value within this field. The actual values to be utilized within the traffic class field are presently undetermined. It shows great potential for voice and video transmission service though. It is not decided as of yet, but the field would probably be set by upper-layer protocols, such as the voice or video applications. [Ref 23:p. 79]

The IPv4 protocol permits all datagram packets to find their own individual paths from source to destination. In the past this has been the beauty of internet functionality. This IP technology allowed for instantaneous changes within the existing worldwide architectures; no individual computer would prevent the populace from receiving and sending message traffic via the internet. Yet, in order for each packet to find its own path each intermediate node must process the packet's entire header to establish the route path for next node delivery. Time requirements for the transmission of entire groups of datagram packets due to VoIP or video teleconferencing demand an evolution of the IP technology. Taking this into account IPv6 offers just such an evolutionary change.

IPv6 appends the individualistic packet delivery process of IP with a header field identified as the flow label. The flow label header field utilizes a 20-bit value to establish a numerical system associating each datagram packet of the same flow with a same flow label value. All packets with a value in the flow label field require special handling. Once the processing node determines the packet is of the same flow as a previous packet it passes the packet along to the same node without the additional requirement to process the remaining portion of the datagram packet header for next node delivery information. The flow concept greatly strengthens the IP configuration by reducing the overhead associated with individual processing requirements as found in IPv4. [Ref 23:p. 78]

C. IPV6 SECURITY CAPABILITY UTILIZING EXTENSION HEADERS

Security is becoming an important requirement for internet communications. It is imperative to individuals and businesses alike that internet traffic is safely delivered to its destination. Often added demands on internet traffic include an assurance of the traffic source to the destination host and safeguarding of the information contained within the transferred data throughout its transit to the destination host. Internet protocol security is examined here as a possible addition to the home communicator.

IPv4 originally developed as a research mechanism between government, education, and business. It then evolved into the production package the world utilizes today. Security although an important aspect of point to point communication was not initially built into the IPv4 package. IPsec was established through RFC 1825 and updated by RFC 2401 to incorporate security into the existing IP architecture. It must be noted that IPsec is not joined to any specific IP version and can be incorporated into both IPv4 and IPv6.

The two means of security for IP technology are the Authentication Header and the Encapsulation Security Header. They can be utilized individually or together. Each supports transport or tunnel modes of operation. Transport mode simply put is a host to host security connection. The tunnel mode involves a security gateway design. Tunnel mode can be utilized with a host node to a security gateway and vice versa as well as security gateway to security gateway. If a security gateway is operating as a host it can utilize the transport mode but only under host status. Authentication and Encapsulation

Security are known as one-way protocols. Both directions must implement these protocols in order to have a two-way functionality. [Ref 23:p. 142]

1. Authentication Header

The Authentication Header (AH) allows the source node to digitally sign outgoing packets. There is no encryption capability with the authentication header. "The Authentication Header provides connectionless integrity, data origin authentication, and an optional anti-replay service." [Ref 24:p. 231] In other words by utilizing the AH the receiver can be confident that the information transmitted is what was intended to be sent and that the sender is who really transmitted the information. Replay attacks are third party intrusions where information intercepted is bombarded to the receiver in hopes of overwhelming the receiving system. The transmitting and receiving hosts must utilize the anti-replay service in order for it to be effective. The transmitting host numbers each packet with a sequence number. The receiving host must check the sequence numbers to be effective.

The authentication header is utilized differently for transport and tunnel modes. With the transport mode the authentication header comes before the next level protocol header but after the IP header and options associated with it. It protects higher level protocols as well as portions of the IP header. IPv4 places the authentication header after the original IP header with all its options. The IPv6 authentication header is placed after hop-by-hop, routing, and fragmentation extension headers. The protection allotted by the authentication in transport mode extends to the payload of the original IP datagram and only parts of the IP header that do not change as they transverse the internet.

In the authentication tunnel mode the original IP header is encapsulated within the authentication header placing it directly behind the authentication header itself. In the case of IPv4 the authentication header comes directly after the new IP header and any possible options required with the new IP header. With IPv6 the authentication header is an extension header of the new IP header. The authentication header, when utilized in tunneling mode protects the original IP datagram in its entirety.

Figure 3-6 displays the authentication header format. The three most significant fields within the authentication header are Security Parameter Index (SPI), Sequence Number, and Authentication Data. The security parameter index is a 32-bit, arbitrary value that uniquely identifies the security association between the authentication header and the destination address. RFC 2401 defines a security association as a simplex (uni-directional) logical connection, created for security purposes. There would be a different security association for authentication header and for Encapsulation Security Payload header as described in the next section. The sequence number is a mandatory counter. The sequence number increments with every datagram transmitted. If the receiving host or gateway monitors the SPI and sequence number, duplicates can be discarded preventing replay attacks. The authentication data field contains the Integrity Check Value (ICV). The security association specifies the authentication algorithm to be employed for the ICV computation. The ICV is computed from IP header fields that are immutable (or predictable at destination), the AH header (with the Authentication Data figured as zero), and upper level protocol data.

2. Encapsulation Security Payload

The Encapsulation Security Payload (ESP) provides authentication as well as encryption to safeguard the payload. It may be used in conjunction with the authentication header or alone. "ESP is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality." [Ref 25] Simply put ESP offers encryption and limited traffic flow confidentiality in addition to the authentication header. Yet, a point to be made is that the authentication found with ESP is not as extensive as found with the authentication header. ESP can be utilized with only encryption, but it is necessary to include authentication either with the ESP protocol or in conjunction with the authentication header. In order to employ the limited traffic flow confidentiality function of ESP it must be operated in the tunnel mode.

```
0                   1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header | Payload Length|            Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Security Parameters Index (SPI)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Sequence Number Field                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Authentication Data (variable)                |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-6 Authentication Header Format

The ESP header is utilized differently for transport and tunnel modes. With the transport mode the ESP header comes before the next level protocol header but after the IP header and options associated with it. Unlike the authentication header in transport mode, when the ESP header is utilized in transport mode it does not protect any portion of the IP header, only upper layer protocols. IPv4 places the ESP header after the original IP header with all its options. The IPv6 ESP header is placed after hop by hop, routing, and fragmentation extension headers. The ESP header can be placed either before or after the destination extension headers, but it only protects the headers that follow it.

In the ESP tunnel mode the original IP header is encapsulated within the ESP header placing it directly behind the ESP header itself. In the case of IPv4 the ESP header comes directly after the new IP header and any possible options required with the new IP header. With IPv6 the ESP header is an extension header of the new IP header. The ESP header, when utilized in tunneling mode protects the original IP datagram in its entirety. The ESP header and its contents are authenticated. The contents minus the ESP header are encrypted.

```
0                 1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Security Parameters Index (SPI)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Sequence Number Field                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Payload Data (variable)                       |
|                                                               |
|                     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     |       Padding (0-255 octets)            |
+-+-+-+-+-+-+-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                | Pad Length    | Next Header   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Authentication Data (variable)                |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-7 Encapsulation Security Payload Header Format

Figure 3-7 illustrates the ESP header format. The ESP Header fields are Security Parameters Index, Sequence Number, Payload Data, Padding, Pad Length, Next Header, Authentication Data. The first two and the last one provide the authentication portion of the ESP similar to what was described above in the Authentication Header. This is an optional function within the ESP header. The remaining four fields correspond to confidentiality within the ESP header. The Payload Data is the information being encapsulated by the ESP header. Its format corresponds to the information found within the Next Header field. The Padding and Pad Length relates to the specific encryption information required for extracting the data. The actual information in these fields depends upon the Security Association selected.

D. IPV6 ADDRESS TO PHONE NUMBER SCHEME

One of the foremost reasons the business world is looking to replace and upgrade IPv4 is the decreasing number of available IPv4 addresses. IPv6 offers a much larger quantity of possible IP addresses simply by quadrupling the IP address space to 128 bits. Another aspect of IPv6 is that it completely alters the present method of address allocation and distribution. The next section describes the address allocation of IPv6 and the following section describes how the new format could be used in such a way to approach the telephone system in use today.

1. Address Allocation/Network Topology

IPv4 addressing notation is represented as a set of four two-digit hexadecimal integers separated by periods or dots. Figure 3-8a displays the IPv4 addressing representation. The hexadecimal values are actually written in decimal format. IPv6 addresses are represented by a set of eight four-digit hexadecimal integers separated by colons. Figure 3-8b shows the IPv6 addressing representation. In order to phase in IPv6 into the worldwide network there is a third format that combines IPv4 and IPv6. This format is represented by a set of six four-digit hexadecimal integers separated by colons followed by four two-digit hexadecimal integers shown in decimal format and separated by periods or dots. A colon separates the first six from the last four values. Figure 3-8c describes the IPv4/IPv6 interim addressing representation. Only existing IPv4 addresses being utilized within an IPv6 environment require this format. The first six values would be 0's and the last four values would be the actual IPv4 address. Subnet masking is shown by a backslash with the number of bits that refers to the prefix. [Ref 23:pp. 91-91]

IPv6 supports three different types of addresses, unicast, multicast, and anycast. Broadcast addresses found with IPv4 are not available with IPv6. IPv6 addresses are hierarchically designed to promote internet routing. The IPv6 format supports private site addresses for organizational use. Also, there are multicast addresses for both local and global usage.

Format	**Example**
a. d.d.d.d	128.253.36.88
(2 digit Hex equals 0-255 possible decimal values)	
b. hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh	FEDC:BA98:0:0:0:0:7654:3210
c. hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:d.d.d.d	0:0:0:0:0:0:128.253.36.88

Figure 3-8 Addressing Representation

The aggregatable global unicast address is given a large block of addresses. Each unicast address identifies a single network interface. If a node has more than one network interface it will have more than one IPv6 unicast address. No two nodes can share a unicast address. Internet unicast addresses have been given the hierarchical structure to facilitate network routing.

Of the entire IPv6 unicast address the first 48-bits make up the public portion. The format of the public portion begins with the format prefix. Figure 3-9 displays the format scheme for IPv6 unicast addressing. Currently the only three-bit prefix code used is "001." 010, 011, 100, 101, and 110 are presently unassigned and may later be utilized for unicast addressing. 000 and 111 are utilized for other address formats. The next block is the TLA ID or top-level aggregation identifier. It is a 13-bit block that is used for the highest-level service providers. The next field is an 8-bit field reserved to expand either the TLA ID or the next field known as the NLA ID or the next-level aggregation identifier. The NLA ID is a 24-bit field to be utilized by the top-level service to identify its subscribers. This can be used to create a subhierarchy or to identify smaller service providers attached to the larger provider.

3	13	8	24	16	64
001	TLA ID	RES	NLA ID	SLA ID	Interface ID

Figure 3-9 Unicast Address format

The last 80-bits of the IPv6 unicast address belongs to the subscriber [Ref 22:p. 799]. The SLA ID or the sit-level aggregation identifier is 16 bits. It can be used to number subnets or to create a subhierarchy of areas and subnets. The final field is the Interface Identifier. Here 64-bits are used to identify the individual device interface. The interface identifiers are based on the IEEE EUI-64 format. This format uses the existing MAC addresses to create the interface identifier. The interface identifiers can be used to globally address each and every network interface uniquely.

Multicast addressing is similar to broadcast technology. The major difference is that IPv6 multicast packets are only delivered to the nodes that specifically subscribe to a particular multicast. The format of a multicast packet is very rigid. Figure 3-10 displays the IPv6 multicast addressing scheme. Multicast addresses can only be used as destination addresses, never source addresses. The first 8 bits designates a multicast address by holding all 1's. The next four bits are individual flags. Only the fourth is presently assigned. If this is 1 it designates the multicast as permanent multicast address, a 0 designates it as a transient or temporary address. The next four bits describe the scope of the multicast group. In other words the value of the scope indicates whether the multicast group can include only nodes on the same local network, same site, same organization, or anywhere within the IPv6 global address space. The final 112 bits designate the group identification. Two different group id's can be the same depending on the scope and the designation of transient or permanent multicast address.

8 bits	4 bits	4 bits	112 bits
11111111	000T	Scope	Group ID

Figure 3-10 Multicast Address format

The anycast address for IPv6 is similar to the multicast. The similarity is that like a multicast address the anycast address corresponds to more than one node. One difference is within the requirement of who exactly receives the multicast packet. All nodes that subscribe to the particular multicast receive the multicast packet but an anycast packet only has to reach one of the nodes designated by the particular anycast. Another difference is that the address format is consistent with a unicast address rather than the multicast address format. Because the anycast addresses are unicast addresses, members of an anycast address must each be explicitly configured to recognize that address as an anycast address. [Ref 23:p. 109]

Finally, for edification there are other IPv6 addresses that are utilized for testing and setup of networks. One such address is the unspecified address or all-zeros address. Computers that have no valid address such as a new computer booting up on an existing network utilize the unspecified address. There is also a loopback address in IPv6. The loopback is a testing address only and is internally used by the node by sending a packet to immediately reenter the node's IP stack but to appear as if it came from an outside source.

2. Regionalize the IP Addresses

The telephone identification scheme is hierarchical by location. When dialing within the same area code one need only dial the last 7 digits, yet if dialing to another country the country code must be dialed before the area code and local number. Looking at IPv6 the unicast address is the point-to-point addressing scheme that would be utilized for all data transfer to and from the home communication system.

The unicast addressing system of IPv6 is also hierarchical in nature but it is hierarchical by top-level service providers not location. One of the major problems that had developed over time with IPv4 addressing was an inability to support internal company networking growth. Table 3-1 shows the address class characteristics of IPv4. If a company had outgrown its class C addressing scheme the additional addressing fields received through ARIN (The American Registry for Internet Numbers), RIPE (The Reseaux IP Europeens), or APNIC (The Asia-Pacific Network Information Center) would not be numerically close to the existing values. Also, it turned out that very few organizations could actually utilize a class A address. This was causing a waste of address space. Today many organizations are utilizing several class C addresses together to because their organization is too small for class B buy too large for a class C address. IPv6's addressing scheme has taken this into consideration and is based on business' internal requirements for addressing and company growth.

Class	Length of Network Address	First Number	Number of Local Addresses
A	1 byte	0-127	16,777,216
B	2 bytes	128-191	65,536
C	3 bytes	192-223	256

Table 3-1 Address Class Characteristics for IPv4

As described above the first three bits for a unicast address is 001. The next field is the TLA ID is 13 bits long and holds a possibility of 8,192 different values. The TLA ID is given to top-level service providers. The top-level service providers will include government and commercial providers. This is against the basic premise of the phone hierarchy in the sense that the country telephone codes and area telephone codes are the same regardless of who the service provider is. Here the service provider will hold the highest order of the numbering criteria. In order to offer a true hierarchical system such as the telephone system the top level service providers would have to maintain the rights to certain wide area locations within the world. This is not feasible.

Since it is not possible to support an overarching hierarchical solution as found with the existing telephone system a next option is to investigate the ability to offer hierarchical capabilities within the top-level service provider down to the customer. The NLA ID is 24 bits long and is the next-level aggregation. The site and/or service provider utilizes the NLA ID to further break down the LAN architecture within an organization. It has the possibility of 16,777,216 values. The SLA ID is 16 bits long and is a site level aggregation. It has a possibility of 65,536 values to allow companies the ability to manage their own hierarchical addresses.

Something worth noting is that the home communicator will be utilized within households rather than companies. The requirement for internal routing within a household is much less than business needs. The internal routing functionality for businesses that is built into IPv6 could be used by the internet service provider. In other words, the top-level service provider could utilize the NLA ID and SLA ID together to offer a greater footprint of service to homes with home communicators. Together the NLA ID and SLA ID are 40 bits offering a total of 1,099,511,627,776 values together. The reserve field in between the TLA ID and NLA ID when used will open up the ability to offer more households for each of top-level service providers. The NLA could be utilized at the country, state, and even county levels. The SLA ID could be utilized at the neighborhood for large cities or town level for the smaller towns. The Interface ID is a 64-bit value based on the IEEE EUI-64 interface ID and will easily identify the different telephones within one household as each will require a separate interface to the network. The NLA and SLA ID's overall could be segregated out by a populous mapping structure throughout the world to offer a similar hierarchical scheme to the telephone numbering scheme.

Unlike the telephone system where dialing local numbers requires fewer digits than long distance the IP technology requires all digits of the IP address to route. This telephone hierarchical scheme could be approached by additional VoIP software. In order to make a long distance telephone call one must dial 1 first. This alerts the telephone switch to the fact it is a long distance phone call. The same could be programmed into the software that changes the telephone input into the VoIP. Yet, because the highest level of IPv6 addressing is the top-level service provider there will always be a requirement to include the top-level addressing code when making a call.

The bottom line is that IPv6 will allow regionalization of IP addresses that approach the existing scheme found with telephone numbers. Yet IPv6 will not equal or surpass the versatile hierarchy technology found with the telephone service unless IANA modifies the proposed format to offer more of a regionalized format. The presently projected organizational format cannot be manipulated to support true telephone scheme technology as long as it requires the top-level service provider at the top of the IP addressing scheme.

E. REQUIREMENTS TO INTEGRATE IPv6 INTO LINUX

Linux unlike other operating system software utilizes the talents of interested participants throughout the world to upgrade and enhance the Linux kernel due to the GNU General Public License it is captured under. The integration of IPv6 into the Linux kernel is no different. In other words consumers do not have to wait for the parent company to establish a new feature such as IPv6 to be able to utilize it. Instead, individuals who have developed modules for IPv6 functionality within the Linux kernel offer them freely on web sites for open testing and modification by other interested participants.

In order to offer the Linux with IPv6 functionality the existing status of IPv6 integration into the released Linux kernels must be validated and lacking functionality must be written into code and tested. Presently the latest stable release of the Linux kernel is 2.1. This version as well as the next, 2.2, has IPv6 functionality built into them. The extent of the Linux IPv6 functionality is captured on the web site: http://www.bieringer.de/linux/IPv6/status/IPv6+Linux-status.html. The pages of this web site shows that certain IPv6 requirements are in process of being developed or still missing.

The incomplete or missing IPv6 functionality required in Linux to support a home communicator includes flow label specific routing (in process), multicast routing (not implemented), security (in process), 6 over 4 (in process), and 6 to 4 (missing). The flow label specific routing of IPv6 (described in section IV. B. 3) will strengthen the telephony capabilities of the home communicator. Multicast routing will expand the home communicator telephone functionality. Multicast will allow the household telephones to operate all on the same connection or independently from each other. The security with IPv6 is additional protection for point-to-point communication. The IPsec, authentication and/or encryption, will supply the consumer with increased safety from telephony or data transfer interception and from outside attacks.

The functionalities “6 over 4” and “6 to 4” deal specifically with the timeframe that IPv6 is sharing the internet with IPv4. Once IPv4 has been replaced completely with IPv6 there will be no requirement for these Linux modules. “6 over 4” is the preparation for an IPv6 packet to be sent through an IPv4 network. A gateway is required at each end to support conversion from IPv4 to IPv6 and vice versa. The home communicator internet service provider would support this capability where required. “6 to 4” is the requirement for an IPv6 packet to be sent on an IPv4 or IPv6 network to an IPv4 host.

Since the telephony capability for the home communicator is directed towards other home communicators with IPv6 or to standard telephones this criteria is not relevant.

Finally the IPv6 addressing scheme must be examined in order to develop a standardized location hierarchy for home communicator sales. In order to achieve this it is important to take into consideration the footprint of expected home communicator sales. Also, future sales locations must be taken into account in order to prevent under utilization of IPv6 addressing capabilities. Once a standard hierarchical scheme has been developed IP addresses can be distributed to customers as each home communicator is sold.

In order to stand up the home communicator system with IPv6 it would be necessary to have a stable Linux kernel that supported the functionality described here. If this functionality were not found within Linux stable kernel releases or separately on Linux support web sites it would have to be developed prior to fielding the home communicator.

F. IPv6 VERSES IPv4

The operational IPv6 internet is known as the 6Bone. "The 6bone is an experimental worldwide network for testing interconnectivity of IPv6 implementations, checking if IPv6 really works well or not in actual situation, and so forth" [Ref 26]. It is expected that eventually the 6Bone will become the world wide internet and IPv4 is ultimately obsoleted. In the interim the only alternative for IPv6 specific traffic is that it be tunneled over or through IPv4 networks. A problem that arises with this is that most of the functionality being developed into the IPv6 packet is lost when it is encapsulated within IPv4 technology.

Increasing this awkward situation, there is no tangible control of the information technology market to assure a speedy conversion to IPv6. Because IPv6 continues to grow at a slow rate the requirement to tunnel through IPv4 remains an obligation for anyone designing networking hardware and software today. This creates is a problem from either standpoint IPv4 or IPv6. All IPv4 technology will have to be replaced with IPv6 equipment when IPv4 is outmoded from the internet. Anything designed with IPv6 technology must wait for a strong IPv6 internet to fully utilize the new protocol functionality to their complete advantage. It is basically a matter of time before the internet becomes IPv6 leaving only patches of outdated IPv4 machinery.

Another related issue is the present state of IPv6 technology. Although Request for Comments have been written describing the utilization of specific protocol aspects there is still a large portion of the IPv6 protocol that has as of yet been undetermined. Although most of these undetermined options are minor to the overall IPv6 scheme some may cause complete redesign of existing IP software to be implemented. These issues will not be fully solved for years after IPv6 becomes prevalent throughout the internet.

On the other hand, internet technology is presently available that utilizes IPv6. In 1994 MCI activated a project known as vBNS (very high performance Backbone Network Service). The project was an internet test bed sponsored by National Science Foundation for university and business research. The vBNS now connects research institutions as well as research institutions that are served by other networks. The vBNS began utilizing IPv6 over its backbone ATM technology as early as July 1998. vBNS has been assigned its own TLA for the 6bone. Although vBNS is offered to educational facilities and students at a lower rate the IPv6 functionality present in the MCI Worldcom 6bone is actually available to all MCI Worldcom vBNS subscribers. [Ref 27] MCI Worldcom has intensions of expanding the 6bone capabilities. In late May 2000 Worldcom announced plans to develop 13 new International Data Centers across Europe within a year. [Ref 28]

G. CHAPTER SUMMARY AND IPv6 CONCLUSIONS

This chapter began with a brief background and description of IPv4. An explanation of the IPv4 header format and IPv4 options presently available to with assist data transfers followed. IPv6 was then described in relation to specific enhancements that have been incorporated for added functionality to internet protocol. Such improvements include the simplification of the header format, additional fragmentation operability rules, and added header fields that reduce the overhead of IP packets during intermediate node packet transfer and assist with overall data transfer. IPsec options are then described from both the IPv4 and IPv6 viewpoints respectively. The IPv6 addressing is then investigated for ability to approach a regionalized numbering scheme. Finally, the incorporation of IPv6 into the existing Linux kernel is described.

IPv6 certainly appears to be the next generation of the internet. Not only does IPv6 support the larger IP addressing requirements facing the world it also adds a level of functionality IPv4 does not approach. When IPv4 was developed there was no way to anticipate the audience growing to the household market or the IP packet data growth to voice and video. Although IPv4 can carry voice effectively today the growing market and requirements placed on the internet will undoubtedly hamper IPv4's usefulness in the future. IPv4 is becoming archaic with packet transfer overhead requirements and lack of useful IP addresses. IPv6 cuts down on packet transfer overhead. This is accomplished because inherently IPv6 requires transfer nodes to only process specific information within the IP header and not the entire packet and IPv6 increases the IP address field enormously.

As stated earlier, it is often not any of the governing information technology organizations that create a standard, but the collective customers' actions over a period of time that creates the standards. With this in mind there are no guarantees that IPv6 will emerge as a self-sufficient replacement to IPv4. Even though MCI Worldcom has introduced IPv6 within its network and begun plans to expand into Europe this is not a guarantee for success. Greg Miller of NANOG (Albuquerque NM) presented a brief entitled "IPv6 on vBNS+" on June 12, 2000, found on the website: http://www.vbns.net/presentations/NANOG_IPv6/tsld001.html. The closing slide was titled "Why are we doing this?" with a final bullet stating "Unfortunately, not because customers are asking us for it." Despite this opinion it is the authors best guess that IPv6 will successfully replace IPv4 almost completely within the next 5 years.

Considering the assumption IPv6 will become an internet standard within the next 5 years it is important that the initial design of the home communicator utilize the IPv6 protocol. A majority of the IPv6 functionality has already been developed into the Linux kernel 2.1 and next release 2.2. The remaining aspects of the protocol can be developed into the kernel prior to completion of the home communicator. There no requirement to have the entire IPv6 functionality designed in prior to delivery of the home communicator. Since a high-level design requirement is 5-year life span the major portions required for IPv6 would have to be completed prior to design finalization and product production.

It must be noted that this decision to utilize IPv6 for the home communicator can be handled one of two ways. The communicator itself could be configured to operate as an IPv4/IPv6 gateway. The home communicator would build the data into an IPv6 packet and encapsulate it within an IPv4 datagram packet in order to transverse the IPv4 network that the communicator was attached to. This would require a software change when the communicator was connected directly to an IPv6 network later on. The second option would be that an outside infrastructure would be designed and developed by the internet service provider that will completely support IPv6 interconnectivity. This infrastructure would have to be extensive to accommodate all consumers operating the home communicator. There would be no requirement to upgrade software later since the home communicator is already operating with IPv6 within an IPv6 environment. Obviously, the internet architecture within any given location or region would direct the version of IP actually utilized for a given home communicator.

IV. SPECIFICATION AND HIGH-LEVEL DESIGN OF THE HOME COMMUNICATOR

This chapter describes the home communicator. The first section lists the high-level requirements. The second section lists the individual functions of the home communicator and how they will be achieved. The third section will describe what the customer's view of the home communicator will be. The forth section lists the software high-level design of the home communicator. The final section of this chapter explains the hardware high-level design of the home communicator.

A. HIGH LEVEL REQUIREMENTS

High-level functional requirements for the home communicator listed:

- **Telephony**: The home communicator package must contain telephony capability.
- **Television**: The home communicator package must contain television capability.
- **E-mail**: The home communicator package must contain an e-mail service capability.
- **Internet Browsing**: The home communicator package must contain the capability to browse the world internet.
- **Chat**: The home communicator package must contain chat capability in conjunction with the world internet browser.
- **Simple Text Editor**: The home communicator must contain a simple text editor capability,
- **Connector to peripheral equipment**: The home communicator must contain the capability to connect additional television(s), telephone(s), and/or computer(s) to the home communicator. The additional television(s) and telephone(s), when connected to the home communicator will receive input

from the home communicator. The additional computer(s) connected will be networked with the home communicator upon connection.

High-level quality of service requirements for the home communicator listed:

- **Aesthetic design**: The home communicator must be a contemporary piece that draws the customer to purchase it because it adds beauty and functionality to his or her household.
- **Reliable**: The software within the home communicator package must be hearty enough to prevent system lock up and/or crashes.
- **Inexpensive**: The home communicator should be inexpensive to the middle and low class clientele. The cost should be between 100-200 dollars.
- **Relatively small**: The home communicator must be small enough to fit in any room of the house. The home communicator will have a 15 inch screen and the unit will be all inclusive within the monitor unit.
- **Self-explanatory setup**: The communicator must be self-explanatory to customers throughout the installation, setup, and overall operation process.
- **Easy to increase functionality**: The customer must be able to add new televisions or telephones to the home communicator on their own.
- **Intuitively obvious to operate**: The communicator must be easy to understand for the most computer illiterate customers.
- **Robust for continuous operation**: The home communicator must withstand 24 hour, 7 day per week continuous utilization.
- **Superior functionality**: It is important that all the functions be available at all times. If someone is using the telephone it is important that the television service is not interrupted.
- **5-year life span**: It is important that a home communicators sold today last for at least five years within the customer's household with no requirement for repair or upgrade.

B. INDIVIDUAL FUNCTIONAL SPECIFICATIONS

- **Television**: The home communicator will supply the capability for customers to watch television on the communicator screen as an individual software application. This television software application or television panel can be sized to encompass the entire screen or sized smaller and placed on the functional desktop in order to utilize other computer functions on the home communicator while displaying the television application.

- **Telephone**: The home communicator will have built in telephony capability. A telephone software application will allow the customer to place a call or accept an incoming call by using the trackball mouse and screen selection. The telephone can be used while operating other computer functions on the home communicator. The home communicator will have a built in microphone and speaker in order to utilize the telephony capability from the home communicator.
- **Web Browser**: The web browser capability will allow access to the World Wide Web from the home communicator. Customers will be able to view the web on the home communicator screen without interrupting service to the television or telephone. The customer will have a keyboard and trackball mouse input in order to utilize this web browsing function. The following three specifications for e-mail, chat, and text editor will utilize the same viewing screen, keyboard, and trackball mouse.
- **E-Mail**: E-mail will be available to the customer through the home communicator. E-mail will be displayed on the screen and manipulated by using the keyboard and trackball mouse. There will be a tone associated with each incoming e-mail while the customer has the e-mail system active. The e-mail program will be relatively simple for the customer to learn and use. The e-mail service will be part of the ISP package of the home communicator.

- **Chat**: Chat capability will be included within the home communicator package to allow the customer to contact other close friends or relatives on line. The chat capability allows you to have a directory of persons that you can call up quickly by using an on line message exchange. Instead of calling and talking to someone you can "chat" by first determining that they are presently on line. Then by typing a sentence or so, and striking the enter key it is automatically sent to the other person for his/her viewing and response.
- **Text Editor**: A text editor will be included in the capabilities of the home communicator to assist the customer with document and e-mail writing. This will operate similar to a word processor function.
- **Connector to peripheral equipment**: The home communicator will have connectors that allow customers to attach additional telephone(s), television(s), and computer(s) for added capability. Once plugged into the home communicator these devices will operate as individual items, even though they will be networked together by the home communicator.
    - **Television**: Each peripheral television connected to the home communicator within a household will possess the capability of receiving all available channels distinct from other televisions located within the household including the home communicator itself. Each television will have a number pad on the television itself or on a television remote in order to turn the television on or off as well as to change channels or raise or lower the volume. This peripheral television service will operate independent of the home communicator telephone and computer capability.
    - **Telephone**: Each peripheral telephone within the household will possess the capability to call another phone within the household, neighborhood,

or even long distance simply by utilizing the touch number pad located on the phone. The peripheral telephones will operate just as standard telephones operate. There is no requirement that the other end have a home communicator; the system will be joined to the public telephone service for customers that utilize the public phone system.

- **Computer**: Each peripheral computer within the household will be networked upon connection to the home communicator. This will also allow customers to connect other network items such as printer(s), facsimile machine(s), and/or photo copier(s) in addition to computers.

C. CUSTOMER VIEW OF THE HOME COMMUNICATOR

The Home Communicator is a unique product that combines the individual telephone and television services into one package for the household while adding a web browsing, e-mail, chat, and text editor capability. Section A described the requirements for the product. Section B described exactly what specific capabilities must be available within the home communicator for the customer. This section will explain precisely what the customer should expect to receive when purchasing the home communicator and how the home communicator will operate within the household.

The home communicator is a one-piece device. Simply put the home communicator will appear as a monitor with a keyboard, speaker, microphone, and trackball mouse built in. Figure 4-1 displays a high-level design sketch of the home communicator. The consumer will use the screen and keyboard located at the front and the trackball mouse located on the right side of the home communicator to control the home communicator and accomplish web browsing, e-mail, chat, text editor, and any other computer functions. The consumer will utilize the speaker for the television and the speaker and microphone for telephone. The home communicator will be powered by standard household electricity. There will be a small microphone and speakers located on the front of the home communicator as well to support the television and telephone requirement of the home communicator. The connector for the electric power is at the rear of the home communicator. The connector for the ISP input (optical fiber) is located on the back of the home communicator. Connectors for peripheral television(s), telephone(s), and computer(s) are also located on the rear of the home communicator.

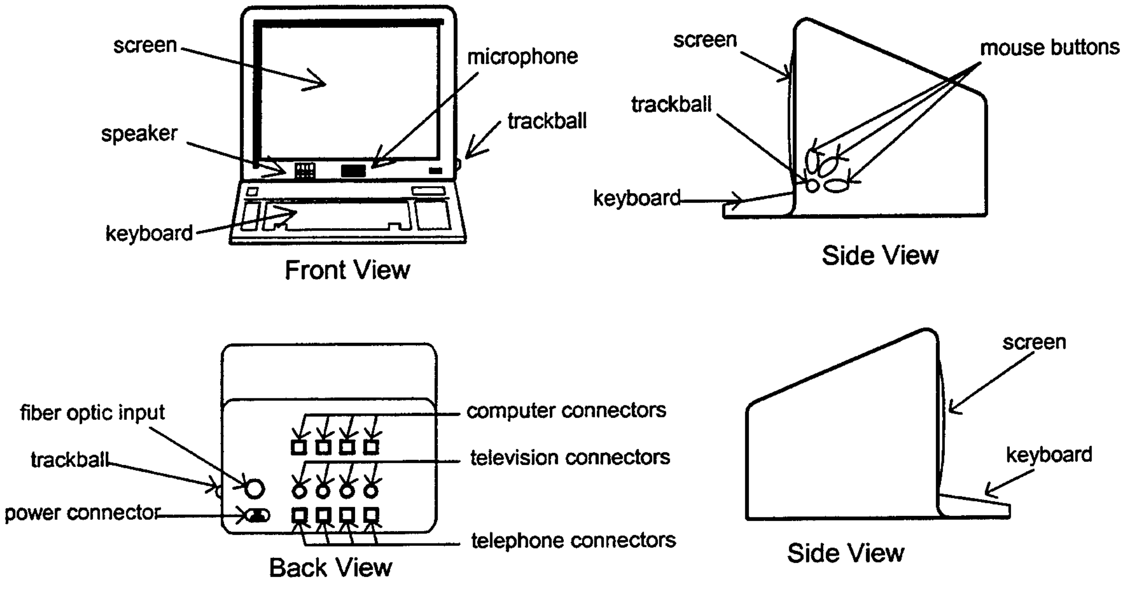


Figure 4-1 Conceptual Views of the Communicator Hub

The home communicator will function as the hub of a home computer network. The consumer may connect any peripheral devices they possess directly to the home communicator completing the household's computer network. These peripheral objects, telephones, televisions, and computers, only need to be standard devices that can be found in any appliance or general purpose store. Figure 4-2 displays how a home communicator system can operate within an ordinary household with peripheral telephones, televisions, and computers.

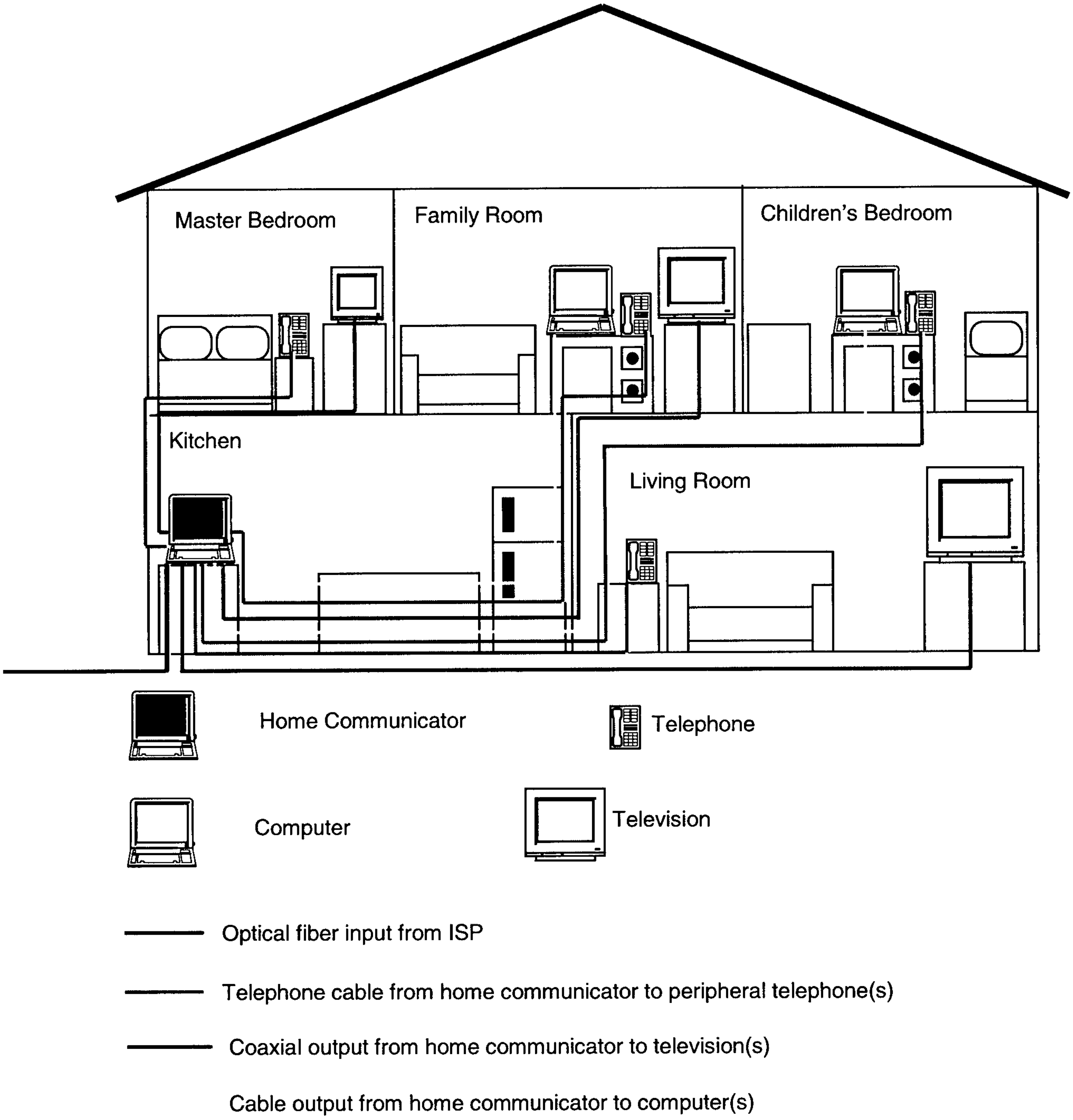
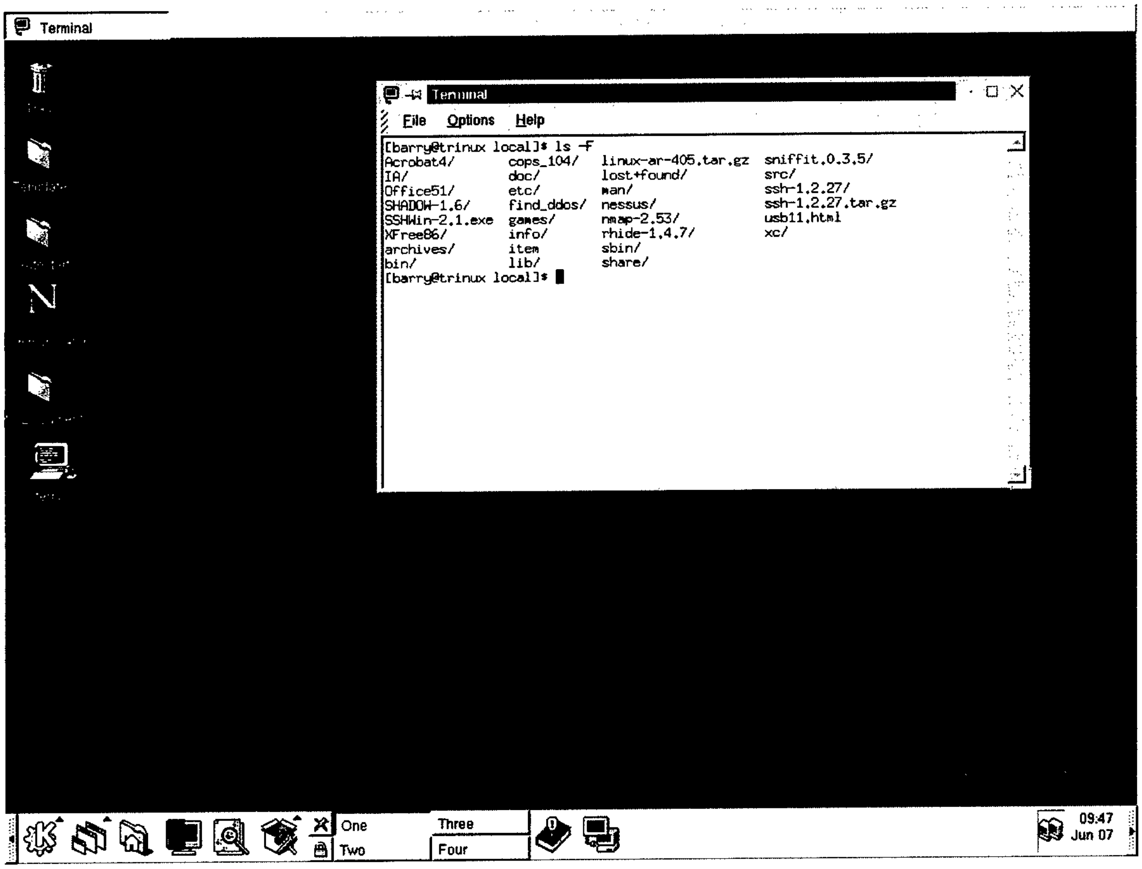

Master Bedroom
Family Room
Children's Bedroom
Kitchen
Living Room
Home Communicator
Telephone
Computer
Television
Optical fiber input from ISP
Telephone cable from home communicator to peripheral telephone(s)
Coaxial output from home communicator to television(s)
Cable output from home communicator to computer(s)

Figure 4-2 Future Household with Home Communicator and Optical Fiber Network. This home has one home communicator, four peripheral telephones, three peripheral televisions, and two peripheral computers. The kitchen has a television and telephone within the home communicator.

The home communicator screen will utilize a simple GUI for the customer's convenience. The intent is to offer a visual selection desktop that is intuitive to any non-computer person. The consumer can select a requested service by using the trackball mouse on the side of the home communicator to place the arrow over the icon and click to open the application. Another method that can be utilized to activate an application will be buttons on the upper or lower Linux KDE panel. The functions available on the home communicator are web browsing, e-mail, chat, text editor, preference selection for peripheral television(s) and telephone(s), and an ability to access any peripheral computers connected through the home communicator. Figure 4-3 depicts a basic view of the home communicator screen displaying possible selection options.



```
[barry@trinux local]$ ls -F
Acrobat4/        cops_104/    linux-ar-405.tar.gz  sniffit.0.3.5/
IA/              doc/         lost+found/          src/
Office51/        etc/         man/                 ssh-1.2.27/
SHADOW-1.6/      find_ddos/   nessus/              ssh-1.2.27.tar.gz
SSHWin-2.1.exe   games/       nmap-2.53/           usb11.html
XFree86/         info/        rhide-1.4.7/         xc/
archives/        item         sbin/
bin/             lib/         share/
[barry@trinux local]$ █
```

Figure 4-3 Basic view of the communicator hub screen

D. SOFTWARE HIGH LEVEL DESIGN

The functionality of the home communicator cannot be accomplished without a complete software and hardware solution. Linux is the core of the software system but without certain application software specific requirements will not be met. Since the choice has been made to use Linux for the operating system it is important to choose software applications that are fully Linux compatible. KDE desktop environment will be the overarching Linux GUI and desktop solution.

The following bullets list and describe the software selections supporting the functional specifications:

- Television: Requires any off-the-shelf software application that would allow the consumer to view an incoming television signal within a computer panel.
- Telephone: The software required to support the telephone functionality most likely would be an off-the-shelf item as well. The telephone software may need to be slightly changed to support the exact requirements of the home communicator.
- Web Browser: The web browser software will be the Linux version of Netscape. The KDE desktop and lower panel will be set up with Netscape icons for ease of customer use.
- E-mail: The KDE version of e-mail (kmail) will be utilized to supply the e-mail capability on the home communicator. The home communicator must be on 7 days per week, 24 hours a day therefore the home communicator will be set up as a mail server. This would preclude the ISP from servicing mail storage servers at intermediate locations. Upon initial set up of each home

communicator it is necessary to default mail storage to a finite time frame to prevent memory space problems on the home communicator.

- Chat: The chat functionality (Chat Client) will also be handled through the KDE desktop software. Chat functionality will also be loaded with Netscape (Instant Messenger) since it is included standard with the Netscape install. The customer will be able to choose whichever he or she prefers to utilize.
- Simple Text Editor: The text editor functionality is another function that is standard within the KDE Desktop environment (Text Editor). As this functionality is quite good for normal requirements this will suffice for the home communicator. Star Office is another possible option but not recommended as its space and RAM requirements will only create a need for greater hardware requirements.
- Peripheral devices require the following software:
    - Television: Requires no software application to support the peripheral television capability. The home communicator will use hardware to supply the constant television signals on to the peripheral televisions.
    - Telephone: The software required to support the peripheral telephone functionality most likely would not be an off-the-shelf item. Although voice has been translated to IP packets with software such as Speak Freely, there is an additional requirement for a translator function within the home communicator to operate a standard telephone connected to the computer. This would have to be specifically designed from examination of existing software packages. A hardware and software solution would be developed during the design phase of the home communicator.

- Computers: The operating software from the home communicator will support access to peripheral computers from the home communicator.

E. HARDWARE HIGH LEVEL DESIGN

The hardware design will operate with the software package to produce a highly reliable multifunction unit. In order to keep cost savings a priority suggestions to use existing off-the-shelf hardware elements are made wherever possible. The overall hardware design has been split into three groups, home communicator, telephone, and television. The home communicator operates as the brain and it controls the overall functionality of the home network.

The actual household wiring required to support the home communicator concept is beyond the scope of this thesis. The basic assumption is that the optical fiber provides the input signal for the home communicator, peripheral computer(s), peripheral television(s), and peripheral telephone(s) to the entire house. The optical fiber connects directly into the back of the home communicator. All of the peripheral televisions and telephones located within the house connect directly to the back of the home communicator as well. The peripheral televisions utilize coaxial cable, the peripheral telephones use the standard 4-wire telephone cable, and the peripheral computer use standard twisted pair cable. The individual hardware item descriptions within this section have not been verified for actual cost and capability. These descriptions are included here to demonstrate overall feasibility of design concept.

1. Home Communicator

The home communicator controls the overall home network system. The home communicator will be a complete package that houses the monitor, CPU, keyboard, and mouse trackball all within one unit. Figure 4-1 illustrates several conceptual views of such a home communicator. The hardware within the home communicator may include an Intel processor Pentium series whichever is readily available at the time of development. The hard drive for the hub will be at least 500 Meg or larger depending on availability and cost. The Linux kernel will boot and run off of read only flash memory. This will protect the setup from possible administrative mistakes by the customer and memory leaks that happen over time with all GUI operating systems. The hard drive will be for consumer mail and file storage. There will be at least 32 MB RAM in order to support the KDE desktop environment memory requirements. The keyboard and trackball mouse will be designed and manufactured specifically for this purpose. This unit will be loaded and configured with the Linux operating system described in the second chapter. The home communicator will be all encompassing with coaxial plugs (F connectors), telephone plugs (RJ-11), and computer plugs (RJ-45) on the back for ease of hookup and television and telephone connection. The input to the home communicator will be an optical fiber plug (ST Connector) located on the rear of the home communicator. Figure 4-1 demonstrates these input and output plugs at the rear of the home communicator unit.

2. Peripheral Telephone

The standard telephones required to support the peripheral telephone capability will be plugged into the home communicator with standard 4-strand telephone wire. The telephone wire will plug into a standard RJ-11 connector located at the back of the home communicator. There will be several phone connections possible into each home communicator. Figure 4-1 demonstrates the RJ-11 connectors at the rear of the home communicator.

In order for us to utilize a normal telephone connected to the communicator there is a necessity to develop a software and hardware solution that will manage the unique requirements of IP telephony and standard voice over wire through a standard telephone. The software must address dial up, connection procedures and conversion of incoming IP packets to voice and vice versa. The hardware must support the telephone RJ-11 connector in conjunction with a NIC card functionality to support the specifications for the home communicator telephone. Each telephone will plug into an individual "telephone NIC card" to support distinct IP addresses for each peripheral telephone. Individual IP addresses prevent the requirement to incorporate expensive off-the-shelf IP telephone technology. Each telephone to the house would maintain the capability to call any other phone within the house or outside the household.

3. Peripheral Television

The peripheral television will be a basic analog television. The only requirement for the television is that it must have a coaxial connection for a cable input that will connect the home communicator to the television. The coaxial cable will be connected to the standard F connector at the back of the home communicator. The coaxial requirement is utilized in order to draw upon presently available technology. The television signal input into the home communicator will be the same signal that is presently utilized by cable television companies. The signal will be continuously sent on a different fiber optic channel from the ISP, possibly utilizing WDM described in Chapter 1B of this thesis. The home communicator hardware will automatically decrypt the separate channel as a television signal and pass the signal directly on through to the television(s). The home communicator will have several coaxial connections to connect several customer televisions from each home communicator. The customer will control the television by using the controls on the television or a television remote. Figure 4-1 demonstrates the F connectors at the rear of the home communicator.

4. Peripheral Computers

Any standard computers that the customer possesses can be connected directly to the home communicator through the RJ-45 plugs located on the rear of the home communicator. Each RJ-45 connector within the home communicator will incorporate a NIC card to initiate the interface between the home communicator and peripheral computer. Each consumer peripheral computer must have a NIC card to facilitate the connection as well.

V. PLAN OF EXECUTION FOR SIX MONTHS

The plan of execution for the home communicator will be broken down into several aspects of development. The first aspect described here will be a list and description of individual tasks required for the three major phases of development: software/hardware design, prototype, and production. The second aspect described here is the overall team strategy and the personnel requirements to support the team strategy. This includes both the quantity of personnel required to staff the teams and the specific skills necessary to accomplish the required tasks. A third aspect is the time line breakdown. Although this plan of execution does not include actual hardware development it will be assumed into the prototype and production phases.

A. DESCRIPTION OF REQUIRED TASKS

Table 5-1 displays a list of basic tasks required to develop the home communicator. The three phases overlap but also maintain an independence from each other. The hardware/software design phase includes the final selection of software and hardware and the design development leading to the prototype phase. The prototype phase is the actual production of a first unit and the time frame included to test and evaluate the design. Any changes to the design software or hardware made before final production fall under the prototype phase. The production phase portrays the time frame after the first product is sold and represents tasks that will be required throughout the life span of the product.

The first two necessities in the design phase are the requirements documentation and the configuration control plan. These both are crucial for the development of the home communicator. The requirements document establishes the target goal of the home communicator. It maintains the vision of the final product and will not change unless requirements cannot be met as originally planned. The configuration control plan must be developed prior to or at the latest in conjunction with the requirements documentation. The configuration control plan will establish how configuration control is to be maintained throughout the development of the home communicator. It will describe specific information that is required to baseline requirements and the hardware/software design packages.

Design Phase	**Prototype Phase**	**Production Phase**
requirements documentation	purchase hardware	develop documentation for customer
configuration control plan	setup hardware	develop final specifications documentation
finalize telephony design	integration testing	purchase/lease control centers for ISP
finalize e-mail design	modify design	set up help desk
initial hardware/software specifications document		modify design for obsoleted items
software modification for IPv6 Implementation		
software development for telephony		

Table 5-1 Tasks required for development of the home communicator by phases

The next requirement for the design phase is the selection of the telephony and e-mail strategy. Because these two capabilities can be handled in any of several ways to support the requirements document it is important to make high level design decisions prior to the finalization of the initial hardware/software specifications document. After the product design (telephony and e-mail) is finalized decisions will be documented within the initial specifications document. The initial hardware/software specifications document will detail hardware and software modifications and extensions required to produce the home communicator. The modification of Linux and application software will then be undertaken.

At the same time the modifications and development of the software packages are taking place the prototype phase begins with the purchase of the hardware. During the beginning of this phase hardware engineers are purchasing material and building the prototype hardware package. After the setup of hardware integration testing will be accomplished to test the feasibility of design. If any problems arise the design will be modified and retested. Any major changes required due to serious testing failure will require a revisit to the design phase in order to evaluate the problem and redesign the prototype.

The production phase begins during the secondary stages of the prototype phase. Customer documentation development is begun at the onset of integration testing if not sooner. In addition to customer documentation the upkeep of the requirements and configuration documentation described above is undertaken during the production phase. The final specifications documentation is undertaken after a successful prototype system is completed. Also during this phase a help desk is stood up for customer support. Any ISP required facilities will be established during this phase as well. Finally, there is an ongoing requirement to monitor for obsolete production items that begins during the production phase.

B. TEAM STRATEGY AND PERSONNEL REQUIREMENTS WITHIN THE TEAMS TO ACCOMPLISH TASKING

The overall project will consist of task-oriented teams. There will be a total of five teams: hardware, software, documentation, prototype, and production. Generally, the software team will be responsible for development of the Linux operating system and application support package. The hardware team will be responsible for selection of Linux compatible hardware that supports the home communicator specifications. The documentation team will handle the requirements, configuration control, specifications, and customer documentation. The prototype team will control all aspects of the prototype process. The production team will be responsible for all items of maintenance occurring after the prototype phase.

Prior to the development of the home communicator the hardware and software team will together be responsible for the major design decisions at the same time the documentation team is responsible for documenting the major design decisions. The prototype team will develop the prototype and report testing results back to the initial hardware and software teams. The prototype team will consist of select members from the software and hardware teams as well as a single software test engineer. This software test engineer will be involved at the beginning of the project only as an impartial consultant; he/she is to remain separate from the design phase itself. This is necessary in order to maintain a level of neutrality to the design during prototype testing.

Any major problems arising during the prototyping phase from hardware and software incompatibility will be resolved through a coordinated effort from the software, hardware, and prototype teams. The documentation team will develop first the requirements and design documentation and later during the prototype and production phase the customer documentation. The documentation team will include select hardware and software team members to assist with the establishment of the home communicator requirements.

The production team takes over from the hardware and software teams after the successful test and evaluation of the home communicator prototype. The production team is made up of hardware, software, prototype, and documentation personnel. The production team will become the customer help desk. The production personnel will also continue to design changes to the home communicator as obsolescence problems arise. The production personnel will assist with high-level requirements such as design and deployment of any internet service provider stations required.

The general experience required for all team personnel to complete the tasking includes Linux operating system, computer networking (TCP/IP), e-mail administration, computer hardware familiarity, and Linux software programming. The personnel required for all the individual teams must have an expert level of experience in at least one or more of these categories to support the project.

In order to set up the five teams as described all persons hired are by necessity required to have several talents. The teams will each consist of anywhere from 4 to 6 members, but as stated above most of the personnel will be on more than one team. At least two team members for the hardware and software teams must also become part of the prototype team. At least one of the hardware and software team members must also become part of the documentation team. The documentation team will consist of 4 members only. The other two of the members will remain on the documentation team only and only assist other teams if imperative. The members hired for the hardware and software teams will either direct themselves into the documentation or prototyping arena when not directly focused on the planning and design sessions. This practice will help maintain a natural level of task management within the company. Also, this practice promotes the employees a high level of information sharing throughout the process.

Approximately fifteen personnel must be hired to support this type business structure. When hiring the individual team members it is important to choose hardware personnel that possess a strong background and interest in technical writing (1 or 2 personnel) or test engineering (1 or 2 personnel). The test engineering experience will directly support the prototype phase. The software personnel hired must also possess a strong background and interest in technical writing (1 or 2 personnel) or test engineering (1 or 2 personnel). It is also important that at least one hardware person has formidable experience with Linux software. The leaders of each group will be chosen from their level of knowledge and management experience. The software test engineer must have specific manufacturing experience from a hardware and software standpoint.

C. TIMELINE OF 6 MONTH EXECUTION

Figure 5-1 displays the estimated timeline for the project. The start dates are given as weeks from the initial start of the project. The time frame column describes the amount of time the evolution is projected to take with all team members present. Although the dates are given as factual it must be understood that flexibility would be important for such an undertaking. There is no way to anticipate hardware and software incompatibilities prior to the actual prototype phase.

TASK	start date	end date	time frame
Design Phase	**wk 1**	**wk 16**	**16 wks**
requirements document	wk 1	wk 3	2 wks
configuration control plan	wk 1	wk 3	2 wks
choose telephony design	wk 3	wk 5	2 wks
choose e-mail design	wk 3	wk 5	2 wks
initial hardware/software specifications document	wk 3	wk 8	5 wks
software modification for IPv6 Implementation	wk 8	wk 16	8 wks
software design for telephony	wk 8	wk 16	8 wks
Prototype Phase	**wk 8**	**wk 26**	**18 wks**
purchase hardware	wk 8	wk 14	6 wks
set up hardware	wk 14	wk 16	2 wks
integration testing	wk 16	wk 20	4 wks
modify design	wk 20	wk 26	6 wks
Production Phase	**wk 8**	**wk 26**	**18 wks**
develop documentation for customer	wk 8	wk 26	18 wks
develop final specifications documentation	wk 8	wk 26	18 wks
purchase/lease control centers for ISP	wk 8	wk 26	18 wks
set up help desk	wk 14	wk 26	12 wks
modify design for obsoleted items	-	-	indef.

Figure 5-1 Estimated Timeline for Home Communicator Development

VI. CONCLUSIONS

Previously all household communication and entertainment services have been viewed as separate items. These include television, telephone, and computer internet services. As computer capability continues to grow a transformation from separate services will migrate into one service. This service operating within a networked household will present television, telephone, web browsing, and computer functionality all from the one incoming signal.

This thesis has shown that overall software presently exists, specifically Linux operating software that could successfully manage such a communications hub. This communications hub would be capable of interpreting a single ISP input to direct the individual services to the proper device. Additional functionality could easily be included through Linux compatible applications offering the customer a simple to operate, stable electronic mechanism. Individual functions offered would include television, telephone, internet web browser, chat, and simple text editor. In order to accomplish this concept, though, there are several major issues to address.

The use of IP telephony poses an unusual problem to the home communicator. It cannot be solved easily through the home communicator design. Software packages presently available offer computer-to-computer functionality or computer to telephone functionality. Present technology requires that in order to reach existing telephone service customers as well as other home communicator customers it is necessary to develop a gateway architecture system connecting into the existing telephone system. The ISP must accomplish this. A final option outside the scope of this thesis would be to design the home communicator as a server that supported direct IP telephony through extensive software design.

Another problem area discovered was the e-mail capability of the home communicator. In order to support e-mail clients the requirement for mail server storage would be shifted to the ISP. Mail server storage would have to be supported either through the IPS or another service. An option outside the scope of this thesis was the design of the home communicator to be a mail server as well as the mail client. This would not be a helpful design considering the increased level of processing required on the home communicator. This additional server software could easily make the communicator too complicated. This would be a detriment since the home communicator is to be marketed as an easy to use mechanism.

IPv6 was investigated for feasibility with the home communicator and found to be acceptable as the internet protocol for such a home communicator. IPv6 proposes less overhead processing by utilizing a smaller header and a different procedure for handling packets along the datagram path. IPv6 offers several new header fields including traffic class and flow label that enhance the IP telephony and large data transfer requirements such as with video teleconferencing. Although IPv6 is not utilized in the main stream of the internet yet there is a strong push to progress towards it before 2010 when the IPv4 addresses run out. Also, new versions of the Linux kernel support various aspects of IPv6 already. Utilization of IPv6 now with ISP IPv6 server support will prevent a later requirement to upgrade home communicator software for internet protocol.

Linux has proven itself a very stable operating system with very few functional problems. It must be noted though that Linux has its own drawbacks. Linux displays a problem with memory leaks. Although the problem is not as pronounced as found with Microsoft problems leaving a Linux computer on 24 hours per day 7 days per week as required with the home communicator will eventually create a system lock up due to memory leaks. Therefore, it is important that the home communicator owner must systematically reboot his or her machine in order to clear the system of memory leaks before it creates a system crash. Utilization of a flash memory for the system boot up disk space would assist with this, especially after system crashes. The flash memory cannot be written to, so even if the owner crashed the system with his or her lack of knowledge the machine would still boot up clean every time. This problem leaves a potential black mark on the home communicator. No one ever has to reboot their present day telephone due to a memory leak or any other reason. Finally, it must be noted that even though Linux can be delivered to the customer configured it will never be completely intuitive to persons who have never seen a computer screen or those that are familiar with Microsoft operating systems. There will be a learning curve and a help desk will be required by the ISP to answer these basic first time questions.

Computer technology has approached a level that can accommodate the home communicator system described here. As seen major decisions must be addressed and supported by the ISP in order to support all the functionality though. As technology continues to expand the world market this home communicator concept will become more prevalent.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Tolly, K., "The Great Networking Correction: Frames Reaffirmed," *IEEE Internet Computing*, vol. 1, no. 5, pp. 16-23, September-October, 1997.

2. Frazier, H, Johnson, H., "Gigabit Ethernet: From 100 to 1,000 Mbps," *IEEE Internet Computing*, vol. 3, no. 1, pp. 24-31, January-February, 1999.

3. Sahasrabuddhe, L. H., Mukherjee, B., "Light-Trees: Optical Multicasting for Improved Performance in Wavelength-Routed Networks," *IEEE Communications*, vol. 37, no. 2, pp. 67-73, February 1999.

4. Kiniry, J. R., "Wavelength Division Multiplexing: Ultra High Speed Fiber Optics," *IEEE Internet Computing*, vol. 2, no. 2, March-April, 1998.

5. Brull, S. V., Elstrom, P., "At 7 1/2 Cents a Minute, Who Cares if You Can't Hear a Pin Drop?" *Business Week*, December 29, 1997.

6. Port, O., "Fiber Optics: Through a Glass Quickly," *Business Week*, December 7, 1998.

7. Taylor, G., Linux Complete, Sybex, 1999.

8. Beck, M., Bohme, H., Dziadzka, M., Kunitz, U., Magnus, R., Verworner, D., Linux Kernel Internals Second Addition, Addison-Wesley, 1998.

9. Apple Computer Web Site, http://store.apple.com/1-800-MY-APPLE/WebObjects/AppleStore

10. Microsoft Windows Web Site, http://www.microsoft.com/windows2000/guide/server/sysreq/default.asp

11. Microsoft Windows Research Web Site, http://msdn.microsoft.com/downloads/sdks/platform/tpipv6.asp

12. Linux Distribution Web Site, http://www.linux.org/dist/english.html

13. Kofler, M., Linux Installation Configuration Use, Second Edition, Addison-Wesley, 2000.

14. Pfaffenberger, B., Linux Clearly Explained, Morgan Kaufmann, 1999.

15. GNOME User's Guide, http://www.gnome.org/learn/

16. KDE User's Guide, http://www.kde.org/documentation/userguide/index.html

17. Microsoft Internet Explorer, http://www.microsoft.com/Windows/ie/TechInfo/techdocs/unixrevgd.asp

18. Netscape Web Site, http://www.netscape.com/download/selectplatform_11_742.html?cp=dpwcc6

19. Speak Freely, http://www.speakfreely.org/

20. Dial Pad Web Site, http://www.dialpad.com/

21. Cisco Products Web Site, http:www.cisco.com/warp/public/146/pressroom/2000/mar00/ent_032700.htm

22. Feit, S., TCP/IP Signature Edition, McGraw-Hill, 1998.

23. Loshin, P., IPv6 Clearly Explained, Morgan Kaufmann, 1999.

24. Miller, M. A., Implementing IPv6 Second Edition, The M & T IP Library, 2000.

25. Kent, S., Atkinson, R., *RFC 2406: IP Encapsulation Security Payload Header*, The Internet Society, 1998.

26. IPv6 working group, *IPv6: Internet Protocol version 6, frequently asked questions*, WIDE Project, ID: v6faq.html, v 1.6 1997/10/24

27. Very High Performance Backbone Network Service website, http://www.vbnsplus.net/main.html, About vBNS+

28. MCI Worldcom website, *Wordcom Accelerates Rollout of Global Data Center Expansion in Europe*, http://www.wcom.com/, News Room

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, California 93943-5101

3. Dr. Luqi, Code CS/Lq 1
   Software Engineering
   Naval Postgraduate School
   Monterey, California 93943-5100

4. Dr. Bert Lundy, CS/LN 2
   Computer Science Department
   Naval Postgraduate School
   Monterey, California 93943-5100

5. Dr. Bret Michael, Code CS/Mj 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, California 93943-5100

6. Donna Fortin 1
   SPAWAR Systems Center, San Diego, Code D831
   53560 Hull Street
   San Diego, California 92152-5000